University of Groningen

Hydrodynamic Imaging with Artificial Intelligence

Wolf, Ben J.

DOI:
10.33612/diss.117884165

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2020

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Wolf, B. J. (2020). *Hydrodynamic Imaging with Artificial Intelligence: detecting submerged objects at a distance using a 2D-sensitive flow sensor array and neural networks*. [Thesis fully internal (DIV), Bernoulli Institute]. University of Groningen. https://doi.org/10.33612/diss.117884165

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 05-05-2023

BEN J. WOLF

HYDRODYNAMIC IMAGING *with* ARTIFICIAL INTELLIGENCE

Detecting submerged objects at a distance
using a 2D-sensitive flow sensor array and neural networks

COLOPHON

This thesis was typeset with LaTeX, using Robert Slimbach's Minion Pro. The design is based on the *classicthesis* template developed by André Miede and Ivo Pletikosić.

FUNDING

The research featured in this thesis has been partly supported by the Lakhsmi project, funded by the European Union's Horizon 2020 research and innovation program under grant agreement № 635568.

PRINTING

Ridderprint B.V.

ISBN

Printed version: 978-94-034-2348-7
Electronic version: 978-94-034-2347-0

© Ben J. Wolf, Groningen, The Netherlands, 2020

university of groningen

Hydrodynamic Imaging with Artificial Intelligence

Detecting submerged objects at a distance
using a 2D-sensitive flow sensor array and neural networks

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. C. Wijmenga
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Friday 13 March 2020 at 16.15 hours

by

Berend Jan Wolf

born on 17 April 1990
in Stadskanaal

Promotor
Prof. L.R.B. Schomaker

Co-promotor
Dr. S.M. van Netten

Assessment committee
Prof. M.S. Triantafyllou
Prof. H. Jaeger
Prof. J. Engelmann

CONTENTS

1 INTRODUCTION

Ever wondered how fish never seem to bump into the aquarium wall, or how they perceive their surroundings in pitch black caves or in the deep sea? In these cases, fish cannot rely solely on vision or sound cues. Instead they use a different sensation altogether; a sensation we mimic and aim to improve in this thesis.

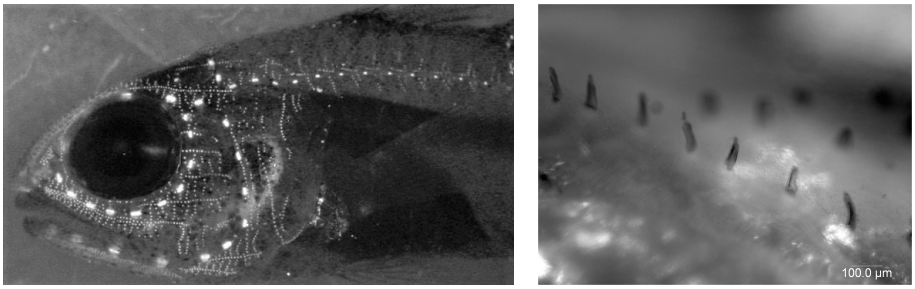
As any paper on the topic will introduce to the reader, fish possess a unique sensing organ, called the *lateral line*, which allows them to detect relative water flow using 1D-sensitive flow sensors. This sensation is often described as *touch at a distance*, as it conceptually rests somewhere in between touch and hearing. Fish use this sensation to perform what is known as *hydrodynamic imaging*: perceiving their immediate surroundings via interpreting the fluid flow as sensed by the lateral line. This allows a wide range of tasks and behaviors, such as detecting objects and other fish, and determining properties of the surrounding fluid flow.

The lateral line has been used as inspiration to create biomimetic implementations of such a sensory system, called *artificial lateral lines*. They are used to demonstrate tasks such as aligning to the flow direction, obstacle avoidance for autonomous underwater vehicles (AUVs), localizing nearby underwater objects, and measuring properties of the fluid flow in pipes or bodies of water.

The scope of the featured research is within the European Lakhsmi project, which aims to develop an underwater sensor system for large scale subsea measurements. In this thesis, we therefore do not necessarily mimic the lateral line, but allow deviations from this design to develop a scalable hydrodynamic imaging solution. We investigate via a series of benchmark tasks whether a novel type of sensor may improve the performance and scale of hydrodynamic imaging.

The benchmark tasks include localizing both vibrating and moving nearby objects, as well as identifying the object shape via the flow measurements of an artificial lateral line. We augment the artificial lateral line by using an array of novel 2D-sensitive flow sensors, which are likely to provide more information than the 1D-sensitive sensitive sensors as found on fish. We use a range of different artificial neural network architectures to assert the performance on these benchmark tasks on different scales, while determining the effect of 2D sensing.

We expand on the topic in the following section, where we further describe the lateral line and hydrodynamic imaging. This is followed by a brief overview of artificial neural networks and their uses in related works. We end by outlining the rest of the thesis and further specifying the research motivation and novelty.



(a) (b)

Figure 1.1: Images of stained neuromasts. (a) *Apogon kominatoensis* with larger fluorescent bright spots indicating the canal neuromasts around the eye, jaw, and the start of a line along the fish trunk. The smaller dots resemble the superficial neuromasts [91]. (b) shows dark stained cupulae of superficial neuromasts of a blind cavefish [86].

1.1 BACKGROUND

Each chapter briefly describes background material on the topic of lateral line sensing and the methods used. We introduce here the common themes throughout the thesis that lead to the research motivation.

1.1.1 *The biological lateral line*

Fish have the unique ability to detect moving and vibrating underwater objects using their mechanosensory lateral line system [36]. In behavioral experiments, the fish lateral line has been shown to be instrumental for several specific behaviors; for instance, prey detection, predator avoidance, schooling behavior, courtship and spawning, rheotaxis (aligning with the flow direction), station holding and spatial orientation [28, 46, 115].

This mechanosensory system consists of several arrays of discrete mechanical sensors positioned along the trunk and the rest of the body of fish. These sensors are called neuromasts, with which fish can perceive the local water motion, or fluid flow, relative to their body.

In the fish lateral line system, there are two types of neuromasts, each with their own beneficial mechanical properties [87, 93]. *Superficial neuromasts* (SN) are present on the surface of the fish body and are in direct contact with the surrounding medium, in this case water. They are mechanically tailored to perceive steady fluid flow speeds, also known as DC flow. The second type of neuromast is called the *canal neuromast* (CNs); they are not in direct contact with the outside flow, but rather are positioned under the fish scales. The CNs are enacted through a relative fluid motion via pores in the canals that house them. The CNs are therefore effectively detecting changes in the fluid flow

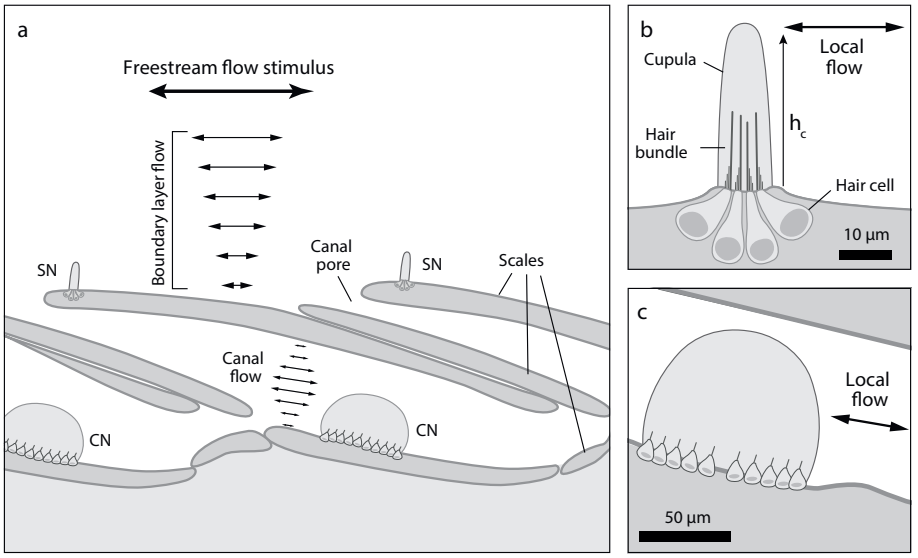


Figure 1.2: Panel (a) shows the positioning and major anatomical features of superficial (SN) and canal (CN) neuromasts. The canal flow is generated by the pressure difference between the pores. Panels (b) and (c) provide a detailed abstraction of the two types of neuromasts. Adapted from [93].

speed, also known as AC flow. These neuromasts are thus mechanically tailored to pick up dynamic perturbations in the surrounding water, enabling fish to sense dipole-like vibratory sources. A schematic overview of the anatomy of the lateral line organ and both superficial and canal neuromasts is depicted in figure 1.2.

A neuromast consists of a gelatinous structure, the *cupula*, which covers several sensory hair cells with protruding hair bundles, also known as *cilia*. When a fluid force is acting on the cupula, the cupula translates or bends, in turn deflecting the hair bundles on each hair cell housed in it. This causes the hair cells to produce action potentials, encoding the measured deflection as a spiking signal.

The anatomy of the neuromast allows this sensory unit to be directionally sensitive. The hair bundles of each hair cell form a stair-case formation. Each of the hairs in this formation is connected using tip links, which makes these hair bundles stiff, and therefore sensitive, in one orientation.

The neuromast as a whole achieves sensitivity along a single axis via combining neuromast with different orientation preferences in a pattern. Hair cells that are sensitive in opposite orientations pair up [41] and are separately innervated by afferent nerve fibers [90], see also figure 1.3a. This results into neuromasts with high directional sensitivity in

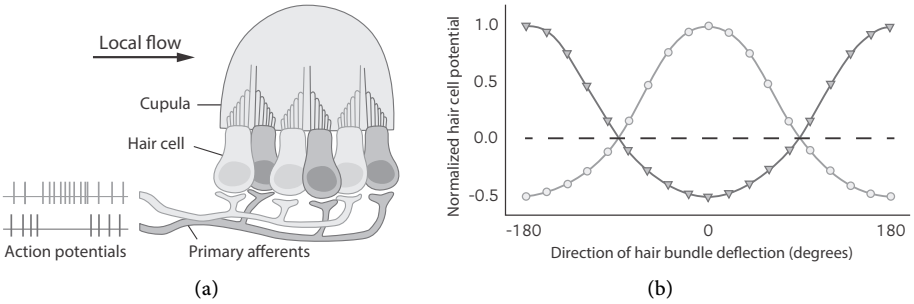


Figure 1.3: Rectified sensing of a neuromast. (a) shows a schematic cross section, where the two populations of hair cells are separately innervated. (b) shows the expected hair cell potential when deflecting the cupula from different directions. Adapted from [18].

one axis (figure 1.3b). The responses from the hair cells in neuromasts are therefore a direct translation of the local flow around the cupula body.

The CNs and SNs differ in their structure, specifically the number of encapsulated hair cells. Where SNs may host typically tens of hair cells, A CN can host hundreds up to a few thousands of these deflection measuring units. This allows the CN to have a bigger differentiation in combined action spike potential, and thus a higher sensitivity to local flow, compensating for the fact that these neuromasts are shielded by the fish scales.

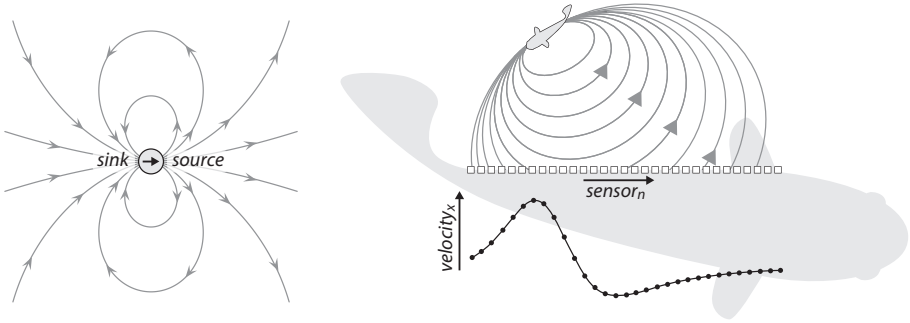
The combined system of the two types of neuromasts provides the fish with perception of the local fluid environment. In the case of the equidistantly spaced CNs along the fish trunk, the spiking patterns of the neuromasts combine to spatio-temporal excitation patterns, which effectively encodes the velocity potential along the trunk [31]. The following section further describes models for fluid flow and how one could use the information contained in an excitation pattern.

1.1.2 *What is hydrodynamic imaging?*

As stated earlier in the introduction, hydrodynamic imaging allows fish to perceive their immediate surroundings via interpreting fluid flow phenomena as sensed by the lateral line.

1.1.2.1 *Hydrodynamic fluid models*

Fluid flow phenomena, be it the progressions of clouds or turbulences in water, can be described via a set of equations called the Navier-Stokes equations [110]. Solutions for the full set of these equations are notoriously hard to compute. Therefore, depending on



(a) (b)

Figure 1.4: Illustrations using the potential flow model. (a) A dipole flow field with visible streamlines. (b) Schematic representation of passive hydrodynamic imaging by the fish lateral line [122].

the situation, some assumptions can be made which simplify the fluid model at the cost of losing descriptive power.

One such simplification of fluid flow leads to an often used model, called *potential flow*, where the effects of viscosity and thus vorticity are ignored [78]. This potential flow is usually introduced in lateral line research via the example of a small vibrating sphere, as depicted in figure 1.4a. When this sphere moves, the water that gets displaced needs somewhere to go; this creates a *source*. Simultaneously, water is attracted towards the low pressure region on the other side of the sphere; creating a *sink*. This causes a flow in the opposite direction: water particles are pushed from one end along a (stream)line around the sphere, and towards the other end. This effectively describes a hydrodynamic dipole, which is an often-used benchmark stimulus for source localization. Such a dipole flow field can be visualized using streamlines, such as in figures 1.4 and 1.6.

1.1.2.2 *Examples of hydrodynamic imaging*

In biomimetic and bioinspired applications, hydrodynamic imaging is a colloquial term for the process of interpreting hydrodynamic measurements to determine properties of objects, obstacles, and the local environment. In the literature, hydrodynamic imaging is used to describe flow direction detection, dipole and object localization, object identification, and obstacle avoidance, or a combination of these tasks. It provides a complementary method to sound based or vision based systems. It is completely passive, does not affect the local environment, and thus enables non-destructive covert sensing.

Conceptually, hydrodynamic imaging can be used in two modes of operation: *active* and *passive*. In the active mode, for example, a fish moves through the water and creates its own flow field; any disturbances in this flow field can be picked up and identified

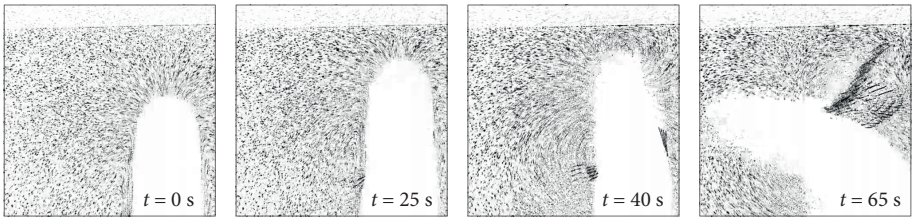


Figure 1.5: Timeline of a fish adjusting its course when approaching a wall. Shown here are streak images made by overlaying five consecutive PIV video frames. Adapted from [119].

as an object such as a fellow fish or as an obstacle such as a rock or wall. In the passive mode, the flow is generated by an external source; this could originate from disturbances and turbulence in streaming water, or from the field generated by nearby moving prey or predators.

Windsor et al. [119] made active hydrodynamic imaging visible via a method called particle image velocimetry (PIV). In figure 1.5 a timeline of a video is depicted which shows the flow field that is generated through the fish's self-motion. At the moment of interaction between the flow field and the wall, the fish stops and adjust its course.

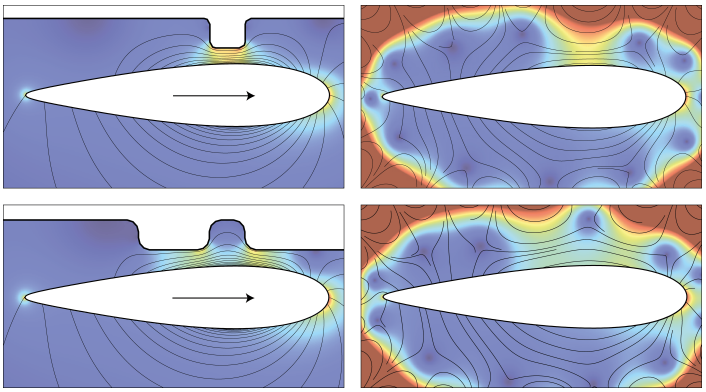


Figure 1.6: Left panes: flow field and streamlines around a simulated fish body shape. The white objects are solids; the color map resembles normalized flow velocity. Right panes: reconstructed flow field and streamlines. Adapted from [117].

Vollmayr et al. [117] provide a method and visualization for reconstructing a flow field based on a simulated lateral line. It is shown in figure 1.6 how a fish body in motion generates their own fluid flow field and how it is affected by key points near a wall. The right panels in this figure show the result of mapping the measured flow field back to an area in the vicinity of the fish body shape. This reconstructed mapping

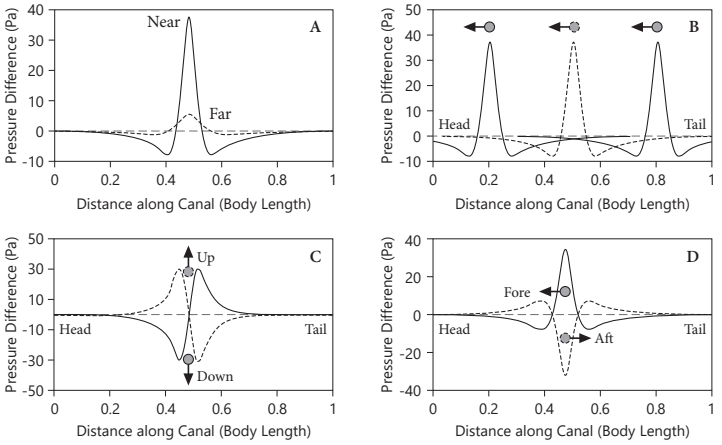


Figure 1.7: Spatial activation patterns. (A) shows the effect of distance. (B) shows the effect of lateral translation. (C, D) show the effect of polarity of movement. Adapted from [29].

clearly resembles the shape of the nearby wall; the reconstructed hydrodynamic image is therefore indicative for solids in the vicinity.

Passive hydrodynamic imaging, and in particular the biophysics of the lateral line, has been studied via several *in vivo* stimulation experiments. The concept behind this passive sensing is illustrated in figure 1.4b, where the smaller fish creates its own flow field, which is then sensed by the larger fish. The velocity profile along the fish encodes the relative location with respect to the lateral line, but can also encode speed, shape, size, and direction of motion.

An early overview of how these velocity profiles change with respect to the location of a vibrating object, also known as a dipole source, was derived by Coombs and Montgomery [29] and is shown here in figure 1.7. The relative distance, location along the array, and direction of motion cause predictable variations of the spatial activation patterns.

Several methods have been devised for localizing a source, based on excitation patterns or velocity profiles. Methods such as the Continuous Wavelet Transform, based on the wavelet nature of the profiles [31], the estimations resulting from spatial descriptors of the curves [43], as well as several other methods including Gauss-Newton [5], beamforming [33, 95], and template matching [125].

A different approach for information processing can be found in another bio-inspired method: the artificial neural network. Several variations of this class of machine learning methods have been used for processing artificial lateral line data to detect, localize, and/or identify objects in water. Before looking into the results of these studies, we first briefly outline general principles and nomenclature for neural networks.

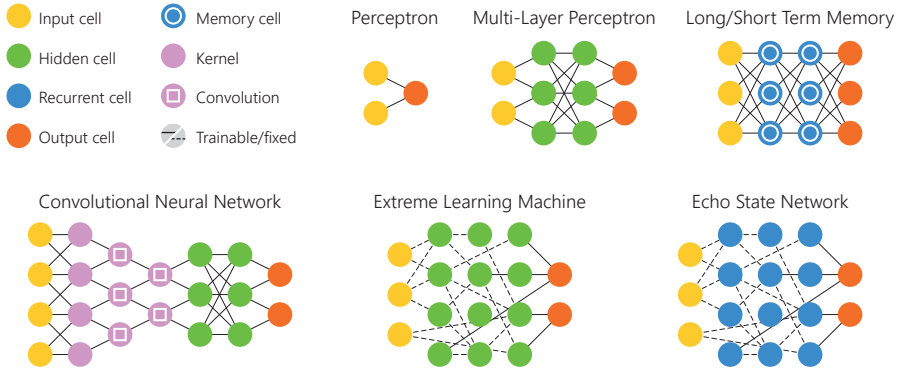


Figure 1.8: Depiction of the artificial neural networks referenced in this thesis. The type of nodes and weighted connections are indicated in the top left corner. Adapted from [116].

1.1.3 *Artificial neural networks in practice*

At their core, neural networks are abstract representations of biological neurons, as found in the central nervous system. Information carrying processes, such as inhibitory and excitatory connections between neurons, the firing rate, and learning are all represented in artificial neural networks. Broadly speaking, all neural networks follow a similar paradigm, where the neurons (or the complete networks) are trained to reproduce a curve or line. This curve is then either used as a regression, mapping input to an expected or desired output value, or it is used as a decision boundary for classification.

The simplest neural network, called the perceptron, creates a single output from multiple inputs. Its output depends on a linear transformation of the inputs, varying the contribution of each input via a set of input weights, usually followed by a non-linear activation function. These perceptrons can be combined into a structure called the multi-layer perceptron (MLP), where in between the inputs and the output perceptron(s), a hidden layer is formed. The hidden layer and non-linear activation functions combined enable the resulting neural network architecture to learn more complex input-output mappings. Moreover, Hornik [62] has proven that the MLP obeys the universal approximation theorem and is therefore a general function approximator.

Other neural network architectures are usually described as variants from this MLP structure. In this thesis, we consider several variants of neural networks. An overview of these network structures can be found in figure 1.8.

1.1.3.1 *Training neural networks*

Neural networks, and their adjective variants such as *deep* and *convolutional*, can be made arbitrarily complex and thus quite powerful. As with any other regression or fitting

method, there exists a risk of overfitting and a risk of underfitting. When reporting neural network performance scores, the relation between training, testing, and validation scores signifies the generalizability of the neural network, i.e. how well the trained system could perform on a similar, but novel data set.

While training an artificial neural network, one needs to find a balance between the available data and the size or complexity of a neural network in terms of trainable parameters, which in most cases are the trainable weights.

If there are more trainable parameters than available examples to learn from, the network risks storing the examples, rather than learning patterns from the data: causing overfitting. Conversely, when the network structure is chosen with too little trainable parameters, it may not be able to capture the underlying mechanics or patterns from the data: causing underfitting. To determine whether either is occurring, one could use a validation set to determine a sensible network structure. This validation set is a subset of the data, and is ideally only used for this optimization step.

The neural network can be trained on a subset of the data set, called the training set, while monitoring the performance on the validation set. Training is the process of iteratively adapting the trainable weights of the network to minimize the error between the current and desired outputs. For the MLP, this is usually done via backpropagation [103, 118], where the error of one or multiple examples is determined, and this error is used to adapt the weights between the output and the previous layer. Once a neural network is trained, the ratio between the score on the training and validation set informs whether the network complexity is adequate.

For reporting the final performance, it is common to use a third set, the testing set, which has not been fed to the network before. The performance of the neural network on the testing set provides a measure how well the model performs on new, unseen data.

To determine whether underfitting or overfitting has occurred, the context of the training and testing performance becomes important. Here, overfitting is signified by the difference in performance on the training and testing set: a considerably lower testing performance signals overfitting. An overfitted trained model does not generalize well to similar but new situations. Underfitting is harder to quantify, since it causes the performance on the training, validation, and testing set to be all lower than optimal. Underfitting and overfitting thus becomes a balancing act between performance and generalizability respectively.

There is a second, more common reporting method to inspect the generalizability of the trained neural network model. The training error is not always reported; more often, the performance is reported using an N-fold cross validation scheme. This involves dividing the whole data set in several subsets, also known as folds, and rotating the role of each set (training, testing, or validation). This allows reporting the testing score statistics, such as the average and standard deviation.

1.1.3.2 *Neural networks for hydrodynamic imaging*

We discuss here the related works in which neural networks were also used to process artificial lateral line data. We discern between the task of classifying and the task of regression, where certain parameters are estimated.

Most works on hydrodynamic imaging with (artificial) lateral lines focus on the localization of objects, which is usually described as a regression problem, determining the relative coordinates and/or direction of motion of a source. The MLP has been used on several occasions for localizing a (dipole) source in a 2D plane [6] or within a 3D volume [131] near the array.

For the second type of neural network predictions, classification, there are some reported examples that used neural networks to classify sources of sensed artificial lateral line data. An MLP was used in all of the following three cases. In [101] the authors describe an experiment in which a sensor array is used to measure three distinct hydrodynamic environments. Via a set of features and a MLP classifier, they were able to discern these three locations based on the hydrodynamic environment. In [82] two differently shaped objects (a circle and square) were moved along an artificial lateral line. From the measured velocity profiles, they were able to discern between the two. Finally, [25] describes the use of an MLP to discern between three modes of von Karman vortex streets, resulting from the interaction between a fish-body shape and an imposed freestream flow.

1.2 RESEARCH MOTIVATION

As evidenced by the previous sections, several simulations and experiments using artificial lateral lines have demonstrated hydrodynamic imaging, on a similar or smaller scale compared to the fish lateral line. The novel contribution of this thesis to the field is summarized in three main themes.

1.2.1 *Scalability*

We investigate whether we can upscale the sensing principle, thus increasing the scale and range of sensing.

While hydrodynamic imaging takes place in the near-field, and thus mainly functions in the vicinity of a flow sensing array, we increase the scale of both array and object. By analyzing the performance of object localization with respect to different distances to the array, we assert the scalability of hydrodynamic imaging.

Closely linked is the choice for the substrate of the system: its physical implementation. If scaling up a sensor array enables large-scale hydrodynamic imaging, the technology should also be scalable itself. We therefore adopt an all-optical sensing technique, which does not directly rely on electricity. It operates passively when connected to an opti-

cal laser emitter and measuring device via optical cables. This is an advantage over (piezo)electric based flow detectors, especially in a marine environment.

1.2.2 *Neural-network based signal processing*

Several methods have been proposed to perform hydrodynamic imaging using flow sensor array data. These range from physics-based methods that infer object properties from the velocity pattern as measured by the ALL, to methods that use a generation method or library of stored examples for their primary information.

We choose to use AI methods, specifically artificial neural networks and feature selection, to train a model for a certain subtask of hydrodynamic imaging. Within this class of machine learning methods, there are many variants of architectures to consider. We investigate in this thesis which neural network architectures are suitable for hydrodynamic imaging.

Since neural networks are powerful estimators, we strive to use neural networks as a tool to find improvements for processing ALL data. By comparing several neural network architectures, we may therefore infer processing steps that can be beneficial to any artificial lateral line system.

1.2.3 *2D-sensitive sensing*

The fish lateral line consists of neuromasts which are only sensitive in one direction. Notably the array of canal neuromasts along the trunk is sensitive in the direction parallel to the lateral line. This may be compensated for by the various arrays of superficial neuromasts, which form intricate patterns of lines as visible in figure 1.1a. Here, arrays of superficial neuromasts seem to form right angles to measure the nearby hydrodynamic environment in 2D.

We investigate whether combining two sensing dimensions into a single sensor may be beneficial for hydrodynamic imaging. Both through theoretical descriptions and several experiments, we assert whether 2D-sensitive sensing has a complementary effect.

1.3 THESIS OUTLINE

This thesis is based on several publications by the author. A list of all publications can be found on page 201. At the start of chapters 2 to 8, we indicate on which paper the chapter is based and specify the contributions of the author via a footnote. We describe here the structure of the rest of the thesis.

In the following chapters, we describe a novel artificial lateral line system which uses neural networks to perform several hydrodynamic imaging benchmark tasks. A graphical abstract is depicted as figure 1.9. We first describe the flow sensing unit,

followed by several simulation studies; these studies investigate whether neural networks are suitable for performing hydrodynamic imaging and whether object localization benefits from 2D sensing. The sensor and neural networks are then combined to test object localization in small and medium scale experiments. In addition, we conduct an experiment to investigate whether the lateral line can be scaled up to several meters and whether it can be used to identify object shapes.

PART I The first part of the thesis comprises chapter 2, describing a fluid-structure interaction model for a 2D-sensitive sensor, bio-inspired by the canal neuromast found on fish. We measure the frequency dependent sensitivity of a sensor prototype, to show to what extent the sensing mechanics scale up for the novel flow sensor.

PART II In the second part of the thesis we feature several simulation studies, starting with chapter 3. In this chapter, we compare three types of regression neural networks for localizing a moving object. using a simulated 1D-sensitive artificial lateral line, we compare the MLP, ESN, and ELM in their ability to reproduce the coordinates of the moving object.

In chapter 4, we forego regression in favor of a heat map approach. We train another type of neural network, the CNN, to reconstruct a heat map indicating the location of multiple objects in a simulated bounded 3D volume. An iterative detection algorithm is presented to assert whether two 1D-sensitive artificial lines are capable of resolving an objects location in 3D.

In chapter 5, we make a direct comparison between a simulated 1D-sensitive and a 2D-sensitive artificial lateral line. Using two localization tasks, we investigate whether there is an advantage to 2D sensing.

PART III The third part of the thesis combines experiments using a prototype 2D-sensitive artificial lateral line with neural network methods to demonstrate object localization. In chapter 6, we extend a theoretical model that describes velocity profiles to apply to 2D-sensing, and investigate whether measurements of a dipole source align with the model. Using an ELM neural network, we demonstrate in a small-scale experiment to what extend a four-sensor array can localize both dipole sources and moving objects.

Chapter 7 describes experiments with an eight-sensor array and two kinds of neural networks, feed-forward (OS-ELM) and recurrent (LSTM). The latter is often used for problems in which the time-dimension is deemed relevant, and has been shown to be more stable than a ESN, which is used in the second chapter. With these medium-scale experiments, we aim to demonstrate localization of a moving object. The (OS-) ELM neural network and LSTM are compared in terms of localization performance, to investigate whether an artificial lateral line system may benefit from recurrent processing.

PART IV The fourth part of the thesis comprises chapter 8, which describes a large-scale experiment, where a sparse eight-sensor array is used to identify nearby moving objects. Specifically, using flow features and an ELM neural network, we investigate what kind of features are informative and whether shape classification benefits from 2D sensing.

PART V The final sections of the thesis include a summary and discussion section, where we summarize the main findings with respect to the research themes and conclude whether 2D-sensing is beneficial for hydrodynamic imaging, what types of neural networks are suitable, and whether hydrodynamic imaging can be scaled up to a supra-biological scale.

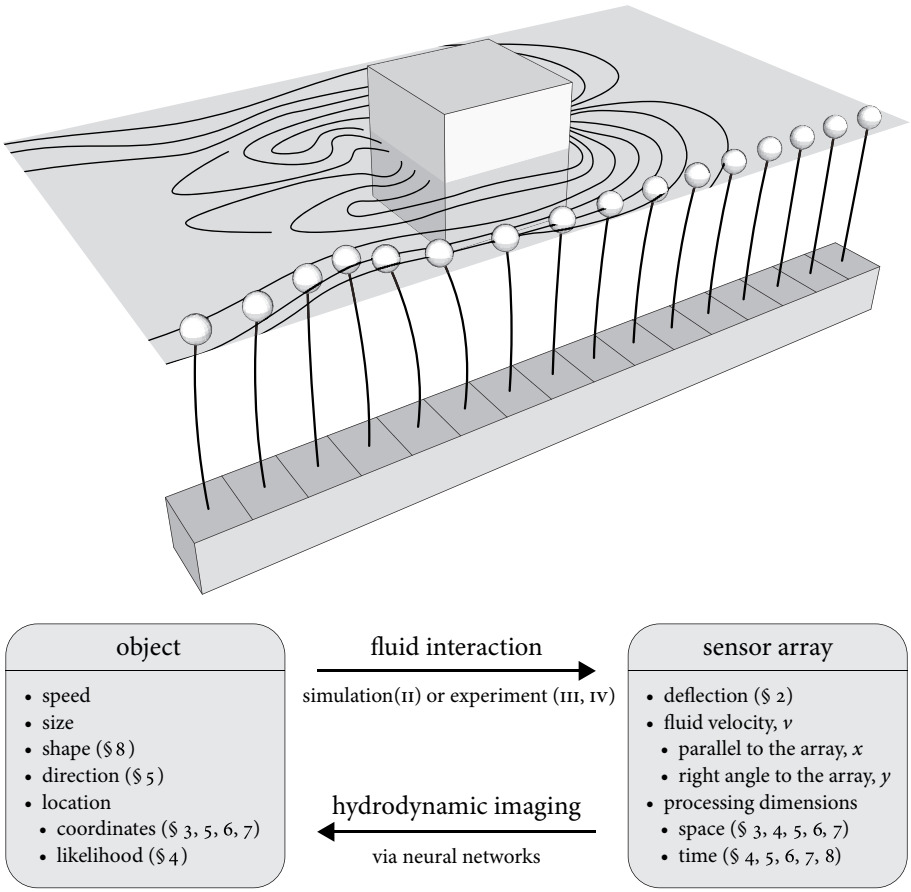


Figure 1.9: Graphical abstract of the thesis.

Part I

SENSING

This part of the thesis describes the development of a novel fluid flow sensor. This sensor is subsequently used for experiments that involve object localization in part III, and object shape recognition in part IV of the thesis.

2

DESIGN AND VALIDATION OF AN ALL-OPTICAL 2D FLOW SENSOR

This chapter describes the development of a novel type of fluid flow sensor, or *artificial neuromast*, bio-inspired by the flow sensing units found under the fish scales: canal neuromasts.

We describe the underlying fluid-structure interaction model of the biological neuromast and adapt this model for an all-optical fluid flow sensor presented here. This novel sensor is made up of optical fibers and makes use of fiber Bragg grating (FBG) sensing. The deflections of this sensor relate to fluid forces and, using the model, we show here how to reliably reconstruct the local 2D fluid velocity vector from the signals from the fiber Bragg Gratings embedded in each sensor unit.

Using a dipole source, we measure the frequency dependent sensitivity of a sensor prototype. This shows to what extent the sensing mechanics scale up for the novel flow sensor and provides a measure for the sensitivity of the sensor itself.

PROVENANCE This chapter has been previously published as:
B.J. Wolf, J.A.S. Morton, W.N. MacPherson, S.M. van Netten, (2018). Bio-inspired all-optical artificial neuromast for 2D flow sensing. *Bioinspiration & Biomimetics* 13(2) 026013.

AUTHOR CONTRIBUTIONS BW conceived the final design of the sensor, performed the experiments, performed the analysis, wrote the draft and wrote the final manuscript. BW and SvN derived the fluid-structure interaction model. JM produced the sensor and revised the draft. WM and SvN conceived the sensing mechanism, supervised the study, and revised the draft.

ABSTRACT The design, fabrication and testing of a novel all-optical 2D flow velocity sensor is presented, inspired by a fish lateral line neuromast. This artificial neuromast consists of optical fibers inscribed with Bragg gratings supporting a fluid force recipient sphere. Its dynamic response is modeled based on the Stokes' solution for unsteady flow around a sphere and found to agree with experimental results. Tunable mechanical resonance is predicted, allowing a deconvolution scheme to accurately retrieve fluid flow speed and direction from sensor readings. The optical artificial neuromast achieves a low frequency threshold flow sensing of 5 mm/s and 5 μm/s at resonance, with a typical linear dynamic range of 38 dB at 100 Hz sampling. Furthermore, the optical artificial neuromast is shown to determine flow direction within a few degrees.

2.1 INTRODUCTION

The design of our optical sensor is directly inspired by a sensory modality used for flow detection by aquatic vertebrates. Fish and amphibians have an array of discrete mechanical sensors at their disposal called neuromasts, which are distributed along the head and trunk. With these neuromasts they can perceive a 1D projection of local fluid motion or flow relative to the body [36]. The biophysical properties of neuromasts, such as the interplay of physiology, mechanics and fluid dynamics, have been studied intensively elsewhere [92, 93].

There are two types of neuromasts, each with their beneficial physical properties to help the fish perceive freestream (DC) and dynamic or oscillatory (AC) flow [18]. Superficial neuromasts (SN) are present on the surface of the body and are in direct contact with the surrounding medium. They are tailored to perceive steady (DC) and low frequency fluid flow velocity. The canal neuromasts (CN) are not in direct contact with the freestream flow, but are housed in internal canals. They are deflected through the pressure difference via flexible membranes or pores in these canals, thereby effectively perceiving freestream (AC) fluid acceleration.

The perceived local fluid flow at each neuromast over time can be concatenated to a spatiotemporal flow pattern, which augments the fish sensory perception [31]. This enables fish to sense fluid perturbations generated by moving sources. In fish behavioral experiments, the lateral line has been shown to be instrumental in many specific behaviors, for instance, prey detection, schooling behavior, and spatial orientation [28, 46, 115].

In order to mimic this biological near-field sensing, several implementations of artificial neuromast sensors have been developed, which also measure a 1D projection of local fluid flow. Some sensors make use of hot wire anemometry [96, 125]. Here, a suspended hot nanowire is cooled down by fluid flow. This links a measurable change in the temperature dependent resistance to fluid flow speed. Most artificial SNs, however, rely on sensing generated strain at the base of a deflecting lamella or cantilever structure in response to fluid flow. This design is favored since the protruding structures escape

unwanted boundary layer effects. Several techniques used for strain sensing include lamella mechanical micro-sensors (MEMs) [38, 86, 126, 127], ionic polymer-metal composites (IPMC) [4, 5, 23] and soft polymer membranes without [11] and with cantilever structures [12, 71, 72]. A technical review of recent contributions to the field of artificial neuromast and artificial lateral lines (ALL) shows an increasing interest in this field [81]. In some cases, orientations of 1D-sensitive sensors are alternated to sense multiple projections, allowing measuring flow perpendicular to the array, e.g. [8, 127].

These SN designs rely on electric methods to operate in wet conditions. Because electric signals are susceptible to noise pickup over large distances, deployment of remote and large scale artificial lateral lines (ALL) has been somewhat limited. We therefore present an all-optical artificial neuromast which aims to address these issues. Our design utilizes fiber Bragg gratings (FBGs), which allows the sensor data to be transmitted through fiber optic cables, thereby enhancing the scalability for all-optical ALLs. Furthermore, our design enables two dimensional fluid flow measurements, thus increasing the information per point measurement.

FBGs are sections in an optical fiber that have been modified to include a periodic variation in the refractive index along the fiber length. This structure reflects light at a specific wavelength that is determined by the spacing of the FBG structure. Stretching or compressing the fiber changes the FBG structure spacing and therefore increases or decreases the reflected wavelength. If the FBG is illuminated with a broadband optical source containing a wide range of wavelength then the reflected (Bragg) wavelength can be interpreted as a function of the applied strain.

Different geometries consisting of two or more optical fibers glued together have been used in order to determine the curvature of the end position of the combined fiber structure [10]. This requires the gratings to be located outside the neutral (bending) axis of a cantilever structure. Examples of FBG curvature sensors [42] and accelerometers [39] have also been developed using multicore optical fibers with multiple cores positioned away from the neutral axis.

Cantilever Bragg grating flow sensing has been used for monitoring steady flow rates [84] and flow perturbations in response to a bluff body [113] in pipes. However, the dynamic properties of the sensors were not examined.

OUTLINE We first present a combined hydrodynamics and strain-structure model which enables sensor characteristics such as its mechanical sensitivity and frequency response to be predicted. Using steady state contact deflection, we infer the linear dynamic range of the sensor. Finally, through hydrodynamically stimulating the sensor at different frequencies, we verify the sensor characteristics and employ a related method to reconstruct flow speeds from sensor readings.

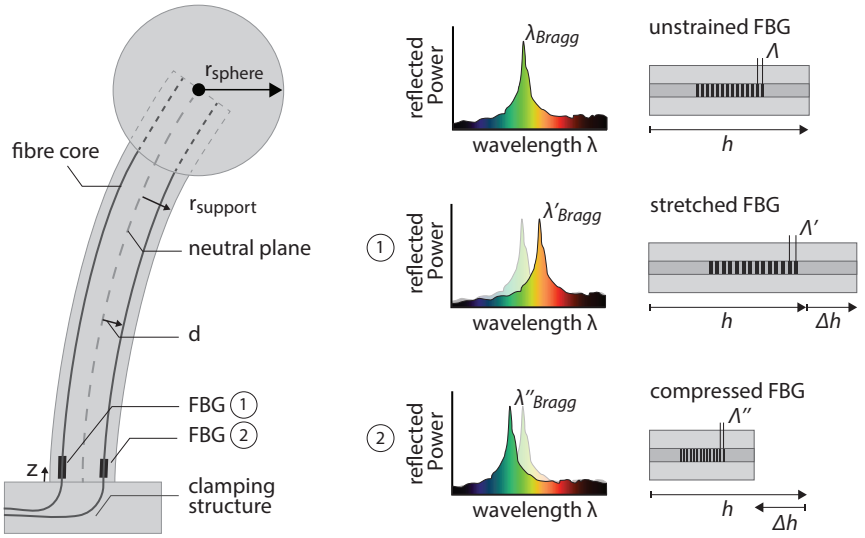


Figure 2.1: Schematic overview of the sensor design and sensing principles. Deflecting the sensor results in local stretching in FBG1 and local compression in FBG2, while the neutral plane doesn't experience either. The reflectance spectra on the right indicate the relation between local compression and stretching, the grating period Λ and resulting detected wavelength peak shift.

2.2 SENSOR AND FLUID MODEL

2.2.1 *Sensor design*

The sensor physically resembles a fluid force recipient spherical body and a fiber support structure providing elastic coupling (figure 2.1). To model the signal of a deflecting sensor, the elastic support is treated as an end-loaded cantilever beam. Using Bernoulli's beam equations, we model the support with length h, having a circular cross section, and possessing flexural stiffness EI (with Young's modulus E and the second moment of area I).

2.2.2 *FBG sensing*

Equations 2.1 and 2.2 show the relationship between the force F applied at a beam's tip, the resulting tip displacement magnitude $\Delta\tau$ and the generated strain ϵ at a distance d from the neutral bending axis at a height z from the fixed cantilever end [16]. The beam is compressed in the bending direction, which decreases strain. On the opposite side of

the bending axis, where the sign of d is opposite, the beam stretches locally, increasing strain.

$$\Delta\tau = \frac{F}{EI}\frac{h^3}{3} \tag{2.1}$$

$$\epsilon = -\Delta\tau \cdot \frac{3 \cdot d \cdot (h - z)}{h^3} \tag{2.2}$$

This strain is measured locally via FBGs (figure 2.1). The FBG comprises a periodic refractive index modulation with a period on the scale of the wavelength of light, such that it reflects light at the Bragg wavelength, λ_B, which is twice the inter-grating distance or grating period Λ.

$$\epsilon = \frac{\Delta h}{h} = \frac{\Delta\Lambda}{\Lambda} = \frac{\Delta\lambda_B}{\lambda_B} \tag{2.3}$$

The Bragg wavelength changes as a function of strain (eq. 2.3) and temperature [59]. The thermal effect can be compensated using differential strain configurations in which the thermal effect is common to all considered gratings [42]. Both the local Bragg wavelength shifts (i.e. sensor signal) and sensor deflections can be practically obtained to ascertain to what extent these are consistent with our present model. Common FBG wavelengths are in the communications band and we choose to work with gratings in this region of $\lambda_B = 1560 \pm 40$ nm.

2.2.3 *Isotropic sensor mechanics*

The support comprises four discrete standard communication SMF-28 fibers, where the light guiding core is centered in a silica outer cladding. The mechanical properties of the fiber support structure are largely determined by the bending stiffness K which is, apart from sensor height, governed by the geometry and composition of the cross section of the elastic support.

Since the support matrix and fibers form a composite material, its flexural stiffness EI depends on the mechanical properties of both materials. The added flexural stiffness of each fiber depends on its squared distance to the neutral bending axis [16]. Although the bending axis may rotate by deflecting the sensor in different directions, we show below that the flexural stiffness—and therefore the bending stiffness K—is independent of a variable bending axis offset α.

In order to show that the flexural stiffness is circular symmetric, only fibers 1 and 2 are considered, since the cross section is mechanically symmetrical above and below the

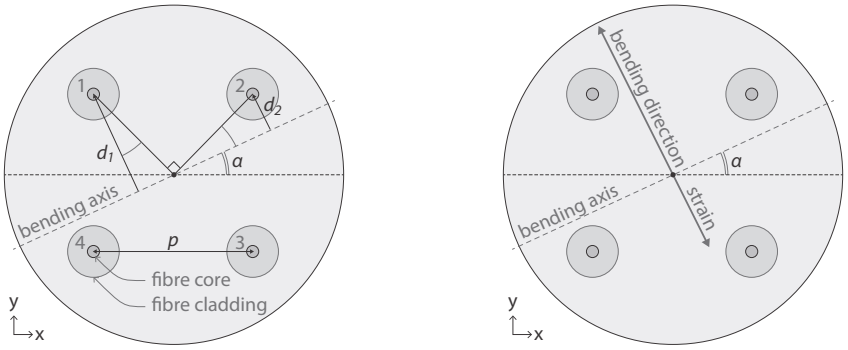


Figure 2.2: Cross sections of the fiber support structure. (a) The schematic representation of the cross section is depicted with a variable bending axis with an angle α relative to the 2×2 square lattice of fibers with side length p. (b) Indication of bending direction and resulting strain projection.

bending axis. When the sensor is deflected over an axis with a variable offset angle α compared to the square lattice, the individual distances d_1, d_2 between the fibers and the bending axis change. But since they form sides of an identical right angle triangle, their summed squared distance to the bending axis ($d_1^2 + d_2^2 = p^2/2$) remains constant and is independent of α. It is only dependent on a fixed p, the distance between two adjacent fiber cores. The flexural stiffness of the support is therefore mechanically circularly symmetric; no preferred bending direction exists.

This results in a direction independent, or isotropic, bending stiffness K determined at the tip of the fiber structure

$$K = \frac{3\pi}{4 \cdot h^3}\Big[E_s \cdot r_s^4 + 4\big(E_c - E_s\big)\Big(r_c^2 \cdot p^2 + r_c^4\Big)\Big], \tag{2.4}$$

where r_c and r_s denote the cladding and support radii and E_c and E_s their respective Young's moduli.

Similarly, in this square lattice configuration (figure 2.2a), we show below that the magnitude of a differential strain vector $\boldsymbol{\delta\epsilon}$, generated by a fixed magnitude of tip deflection $\Delta\tau$, is not affected by a variable bending axis offset α. First, we define the fiber distances:

$$\begin{aligned} d_1 &= \cos(\pi/4 - \alpha)\sqrt{2} \cdot p/2, \\ d_2 &= \sin(\pi/4 - \alpha)\sqrt{2} \cdot p/2, \\ d_3 &= -d_1, \text{ and} \\ d_4 &= -d_2. \end{aligned} \tag{2.5}$$

When deflecting the sensor, with an offset bending axis α, in the direction $\alpha + \pi/2$ and using equation 2.2, the Cartesian differential strain projections then become

$$\begin{aligned}\delta\epsilon_x &= \epsilon_2 - \epsilon_1 \\ &= \mathrm{C}\Big[\sin\Big(\frac{\pi}{4} - \alpha\Big) - \cos\Big(\frac{\pi}{4} - \alpha\Big)\Big] \\ &= \mathrm{C} \cdot -\sqrt{2} \cdot \cos\Big(\alpha - \frac{\pi}{2}\Big),\end{aligned} \tag{2.6}$$

$$\begin{aligned}\delta\epsilon_y &= \epsilon_2 - \epsilon_3 \\ &= \mathrm{C}\Big[\sin\Big(\frac{\pi}{4} - \alpha\Big) + \cos\Big(\frac{\pi}{4} - \alpha\Big)\Big] \\ &= \mathrm{C} \cdot -\sqrt{2} \cdot \sin\Big(\alpha - \frac{\pi}{2}\Big),\end{aligned} \tag{2.7}$$

with

$$\mathrm{C} = \frac{-\Delta\tau \cdot 3 \cdot (h - z) \cdot \sqrt{2} \cdot p}{2 \cdot h^3}, \tag{2.8}$$

where ϵ_n denotes the generated strain at core number n. Then, working out the magnitude and angle of differential strain in polar coordinates, we find that

$$r_{\delta\epsilon} = \Delta\tau \cdot \frac{3 \cdot p \cdot (h - z)}{h^3}, \text{ and} \tag{2.9}$$

$$\theta_{\delta\epsilon} = \alpha - \frac{\pi}{2}. \tag{2.10}$$

The absolute differential strain magnitude $r_{\delta\epsilon}$ shows that we can generalize the use of individual fiber distances d to the inter-fiber distance p in case of differential strain. Furthermore, the direction $\theta_{\delta\epsilon}$ of increased differential strain lags the bending axis by $\pi/2$ and is therefore opposite to the bending direction, which is consistent with equation 2.2.

Since the strain vector $\boldsymbol{\delta\epsilon}$ is opposite to to the bending vector (figure 2.2b), and we aim to measure flow speeds in the bending direction, we choose a pairing of sensor cores such that the projection of differential wavelength shift ($\boldsymbol{\delta\lambda} \propto -\boldsymbol{\delta\epsilon}$) is in the bending direction. With respect to the sensor orientation as depicted in figure 2.2, the Cartesian projections of wavelength difference are given by

$$\delta\lambda_x = c_1 - c_2 = c_4 - c_3 = G \cdot \Delta\tau_x, \tag{2.11}$$

$$\delta\lambda_y = c_3 - c_2 = c_4 - c_1 = G \cdot \Delta\tau_y, \tag{2.12}$$

where c_n denotes the generated Bragg wavelength shift at core number n. Alternative combinations of cores, such as along the diagonals of the lattice, also provide the required orthogonality. This diagonal combination requires four functional cores and is therefore less redundant.

The induced differential wavelength shift $\boldsymbol{\delta\lambda}$ and deflection $\boldsymbol{\Delta\tau}$ can be conveniently be defined as vectors and are thus only a dimensionless linear geometrical factor apart (eq. 2.13).

$$G = \frac{|\boldsymbol{\delta\lambda}|}{|\boldsymbol{\Delta\tau}|} = \frac{3 \cdot p \cdot (h - z) \cdot \lambda_B}{h^3} \tag{2.13}$$

Using practical values for p of tenths of millimeters, combined with sensor height h in the order of centimeters, the value of G is typically in the order of 10^{-7}.

2.2.4 *Dynamic properties*

The physical parameters of both the elastic support and sensor body can be adjusted to predict and obtain desirable dynamic and filter characteristics. We obtain the frequency dependent sensitivity of the sensor by adapting a model for CNs found in fish [92]. The magnitude of the complex frequency response $FR(f)$ is defined as the frequency dependent ratio of sensor response (in this case sensor body motion) per unit fluid velocity. Its argument constitutes the frequency dependent phase lag. In this model, given the relative dimensions of the sensor body and fiber support, we neglect fluid forces acting on the support.

Only three independent physical parameters determine the frequency response: the transition frequency f_t, which selectively shifts a constant shaped Bode plot along the frequency axis, a resonance number N_r, which determines the shape of the associated Bode plot, and the novel buoyancy factor b of the sensor body, which also affects the Bode plot shape. The frequency response and its parameters are described by equations 2.14 to 2.17.

$$FR(f) = \frac{1}{2\pi f_t} \cdot \frac{1 + \frac{\sqrt{2}}{2}(1+i)\left(\frac{f}{f_t}\right)^{\frac{1}{2}} + \frac{1}{3} i \frac{f}{f_t}}{N_r + i\frac{f}{f_t} - \frac{\sqrt{2}}{2}(1-i)\left(\frac{f}{f_t}\right)^{\frac{3}{2}} - \frac{2}{9}\left(b + \frac{1}{2}\right)\left(\frac{f}{f_t}\right)^2} \tag{2.14}$$

$$f_t = \frac{\mu}{2 \cdot \pi \cdot \rho_{fluid} \cdot a^2} \tag{2.15}$$

$$N_r = \frac{K \cdot a \cdot \rho_{fluid}}{6 \cdot \pi \cdot \mu^2} \tag{2.16}$$

$$b = \frac{\rho_{body}}{\rho_{fluid}} \tag{2.17}$$

The sensor is a dampened resonator; its resonant character is parameterized by N_r and affected by the bending stiffness K, sensor body radius a, fluid viscosity μ, fluid density ρ_{fluid}, and body density ρ_{body}.

The resonance frequency f_r (eq. 2.18) indicates the frequency at which the maximum of the FR is located. The resonating behavior is quantified through the quality (Q) factor (eq. 2.19), which only depends on N_r. Keeping the Q factor low causes the sensor low pass filter function (i.e. frequency response) to remain as flat as possible, effectively increasing its usable bandwidth.

$$f_r \cong f_t \cdot \sqrt{N_r \cdot \frac{9}{2} \cdot \frac{1}{b + \frac{1}{2}}} \tag{2.18}$$

$$Q = \frac{f_r}{\Delta f} \cong \sqrt{\frac{2}{3}} \cdot \left(\frac{N_r}{3}\right)^{\frac{1}{4}} \tag{2.19}$$

The bandwidth itself is bounded by a cut-off frequency f_c, the frequency at which the response in the fall-off region matches the DC response and is given by

$$f_c = f_t \cdot N_r \cdot \sqrt{\frac{3}{2(b + \frac{1}{2})}}. \tag{2.20}$$

2.2.5 *Design optimization*

Using the dynamic and mechanical properties, we can optimize some sensor parameters with respect to fluid sensing. With $FR(f = 0)$, the response at low frequencies can be found. Together with the cut-off frequency this leads to a sensitivity-bandwidth (SB) product, which only depends on b:

$$SB = FR(0) \cdot f_c = \frac{1}{2\pi} \cdot \sqrt{\frac{3}{2(b + \frac{1}{2})}}. \tag{2.21}$$

The SB is clearly maximal when $b = 0$, then $SB = \sqrt{3}/(2\pi)$, which is an 73% increase compared to neutrally buoyant neuromasts as found in fish [93]. Using polyethylene spheres with an effective b of about 0.05 in water at room temperature, allows for an increase of SB of about 65% over neutrally buoyant sensors.

Given the geometric factor (eq. 2.13) and the frequency response (eq. 2.18) at low frequencies, we can also optimize the generated differential wavelength shift per fluid velocity by varying the inter-fiber distance p.

$$\frac{|\boldsymbol{\delta\lambda}|}{V} = \frac{24 \cdot a \cdot \mu \cdot p \cdot (h - z) \cdot \lambda_\mathrm{B}}{\left[E_\mathrm{s} r_\mathrm{s}^4 + 4\big(E_\mathrm{c} - E_\mathrm{s}\big)\Big(r_\mathrm{c}^2 p^2 + r_\mathrm{c}^4\Big)\right]} \tag{2.22}$$

In equation 2.22, it is reflected in the numerator that the wavelength shift per fluid velocity increases linearly with the core distance p. However, the optical fibers have to be compressed and stretched at effectively larger distances, adding to the flexural and therefore bending stiffness which is reflected in the denominator with a factor of p^2. As a consequence, p has an optimal value for maximizing detected wavelength shift per fluid velocity at

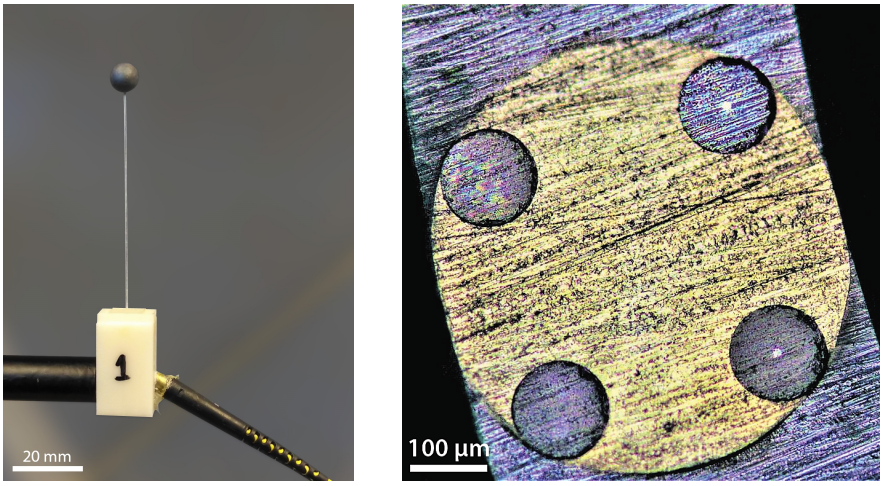
$$p_\mathrm{opt} = \frac{1}{r_\mathrm{c}} \cdot \sqrt{\frac{E_\mathrm{s} \cdot r_\mathrm{s}^4}{4(E_\mathrm{c} - E_\mathrm{s})} + r_\mathrm{c}^4} \tag{2.23}$$

With the chosen materials and related Young's moduli, this optimal distance is smaller than the core diameter. Therefore the optimal design has the fibers as close as possible to each other. Fabrication constraints limited the practical separation of the fibers to a minimum of $p = 0.3$ mm.

2.3 METHODS

In the manufacturing process, four standard communication optical SMF-28 fibers (jackets removed, $r_c = 62.5\,\mu$m, $E_c = 75$ GPa) are suspended in a square column formation. This leaves room for the optical adhesive to flow around the fibers. A custom glue dispenser and UV-curing system is encapsulating the four fibers and is slowly moved along the fiber formation. This embeds the fibers in a matrix ($E_s = 0.14$ GPa) of UV-cured optical adhesive (NOA68, Norland Products Cranbury, NJ, USA). A cross section of the resulting fiber structure is shown in figure 2.3b. The fiber structure is then glued into a mounting block using an epoxy adhesive, in which a small diameter hole acts as a fixed cantilever point for the sensor. The sensor with length $h = 64.8$ mm is fitted with a polyethylene sphere $a = 4.00$ mm, $b = 0.05$.

Figure 2.3b shows a cross section of the tip of the support structure, which is the closest indication of the cross section at the FBG height z. Here, the average distance



(a) (b)

Figure 2.3: (a) The all-optical sensor has a height of 64.8 mm and spherical body radius of 4.00 mm. The fiber structure is glued into a 3D printed block, allowing four separate outgoing optical fibers. (b) Phase contrast image of a polished sample of the cross section of the sensor tip in figure 2.3a. Four fibers are embedded in an adhesive matrix. In the two fibers on the right, the core is made visible by propagating white light in the fibers to illuminate the core.

between the fiber cores is found to be $p = 313.1 \pm 5.4\ \mu$m. The optical fibers do not form a perfect square, therefore we can expect some variation of bending stiffness depending on the bending direction. With the average radius of the support $r_s = 277.4 \pm 5.5\ \mu$m, and the center of the inscribed FBG sections located at $z = 5$ mm, we expect a geometrical factor (eq. 2.13) of $G = 3.22 \pm 0.05 \cdot 10^{-7}$, which describes the relation between the generated differential wavelength shift (sensor signal) per deflection at height h. On the basis of the dynamic properties described in section 2.2.4, we expect the sensor to have a resonance frequency of 13.4 Hz and a Q factor of 13.4.

First, we validated that the sensor response is linear in our operational range by mechanically deflecting the sensor. This allows for a linear relation between sensor body motion and resulting optical signals via a geometric factor, and is a requirement for acquiring the frequency response FR using the current method. Using a linear stage, the sensor was deflected in steps of 0.5 mm while monitoring the sensor signal. Then, in order to infer its dynamic properties, i.e. the FR, the sensor motion and sensor signal were monitored in response to a hydrodynamic dipole stimulus (figure 2.4).

Sensor body motion was measured with a calibrated Zeiss Axiotron microscope. A 2D position sensitive detector (On-Trak PSM 2-2) allowed for high-speed 2D tracking

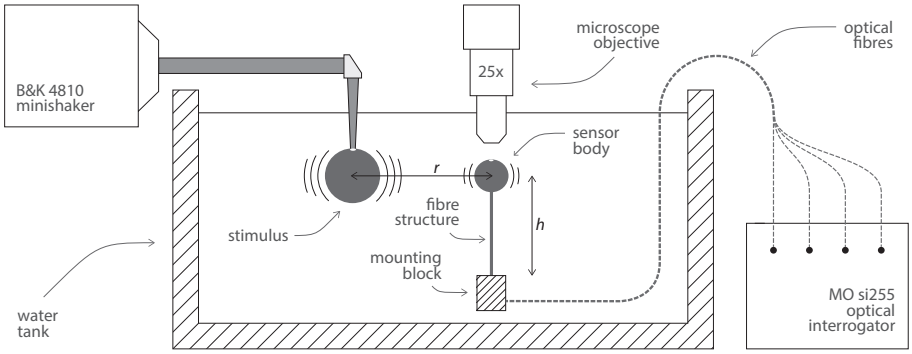


Figure 2.4: Schematic side view of the setup. The submerged stimulus sphere vibrates with a calibrated motion amplitude of about 58 μm at a distance r to the spherical sensor body. The optical fibers (dashed grey lines) are individually connected to optical interrogator channels to measure wavelength peak shifts.

of an attached reflectance marker within a range of 500 μm × 500 μm in the objective focal plane. Strain-induced FBG wavelength peak shift was measured with picometer resolution at 5 kHz using an optical interrogator (Micron Optics si225, Atlanta, USA). The difference in reflectance peaks of cores 1 and 2 and those of core 4 and 1 (figure 2.2a) are used as Cartesian x and y projections of measured wavelength shifts (Eq. 2.11, 2.12).

A hydrodynamic dipole source was produced using a Bruel & Kjaer 4810 mini-shaker driving a submerged sphere ($\oslash$ = 9.9 mm). This stimulus was leveled with the sensor height h. The water tank setup (figure 2.4) was placed on a vibration isolation table (Newport VW series) as to avoid mechanical noise. Sinusoidal stimuli were generated, and sensor body motion synchronously acquired, using a CED power1401 data acquisition system. The mini-shaker was calibrated for frequencies ranging from 1 to 200 Hz, with a constant travel amplitude of about 58 μm. A model for viscous flow [92] was used for calculating the fluid flow velocity produced by the calibrated stimulus. The stimulus sphere was positioned at a distance r, measured using a micrometer stage.

The sensor was pre-stimulated at a given stimulation frequency for at least 3 seconds before sampling, in order to avoid transients. For both sensor motion and sensor signal, we obtained the amplitude and phase lag by applying a flat-top window suitable for low amplitude and resolution data [57], and calculating the discrete Fourier transform. From the resulting spectrum magnitude, we take the maximal value at the stimulation frequency as the response amplitude. The corresponding imaginary part yields the phase-lag at that frequency. In order to prevent spectral aliasing in Fourier analysis, 64 low pass filtered periods of sensor motion with 512 samples per period were sampled for a given stimulation frequency.

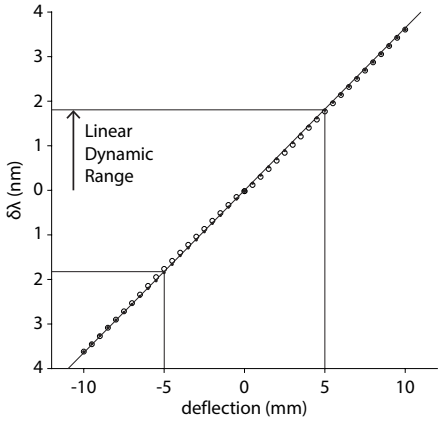


Figure 2.5: Averaged sensor output in response to deflection. The black points show the steps away from the resting position, the white points show the steps from the maximal deflection back to the resting position.

2.4 RESULTS

The Cartesian (x, y) sensor signals for a sensor at rest showed normal distributed noise on the measured differential wavelengths with a standard deviation of 2.02 pm at 5 kHz sampling, which defines a lower bound for the dynamic range of the sensor.

2.4.1 *Linearity*

To test for linearity, the sensor has been deflected in steps of 0.5 mm up to 10 mm (black) and back to origin (white) for both positive and negative deflections, where each position was held for at least four seconds.

From figure 2.5, we infer that the sensor response is linear for deflections up to 5 mm as a conservative estimate ($R^2 > 0.999$), and approximately linear up to 10 mm with some small deviations. On average, a deflection of 5 mm corresponds to a differential wavelength shift of 1.78 nm. The slope of the response matching this linear part yields a measured geometric factor (eq. 2.13) of $3.56 \cdot 10^{-7}$ under static deflection conditions.

2.4.2 *Dynamic response*

To determine the frequency response and further verify the geometric factor at dynamic millimeter deflections, the sensor was hydrodynamically stimulated at frequencies from 2 to 100 Hz. Figure 2.6 shows the frequency response for the sensor body motion in the x and y directions.

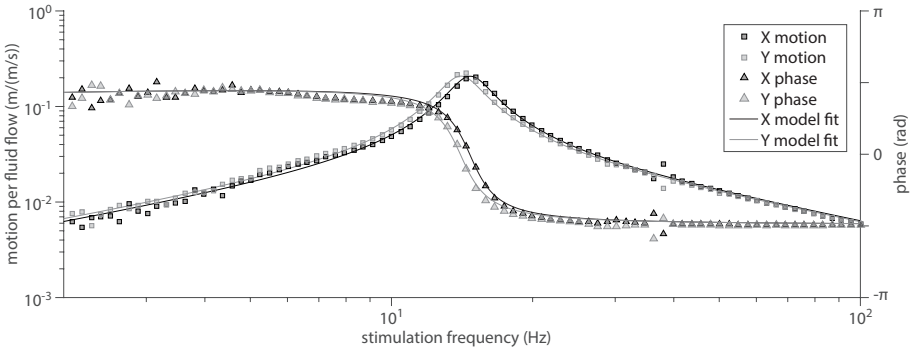


Figure 2.6: Measured frequency response and model fit for both x and y projections.

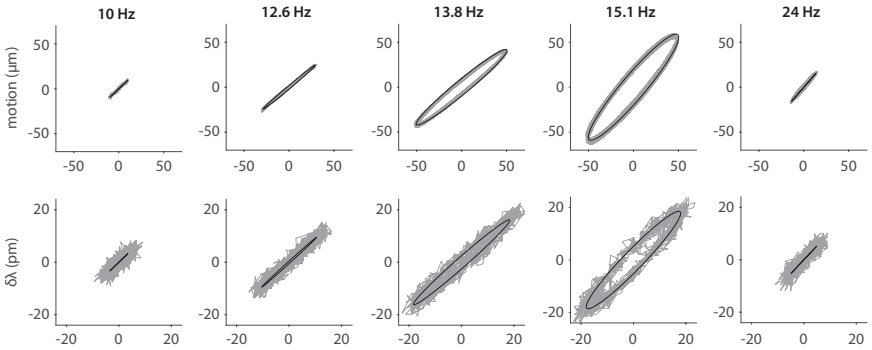


Figure 2.7: Comparison between sensor body motion (top) and differential wavelength shift (bottom). For both types of data, the black ellipses are reconstructed from the flat-top windowed DFT amplitude and phase estimation on the raw (grey) data.

Here, the x and y projections of measured sensor body motion (squares) and phase (triangles) are plotted for each stimulation frequency. Their respective fits result in the fitted parameters $b = 0.05$, $f_t = 0.021$, $N_{r,x} = 7.12 \cdot 10^4$ and $N_{r,y} = 6.52 \cdot 10^4$.

The measured frequency response (figure 2.6) shows a slight difference in resonance peaks and phase lag for the x and y directions. From this measurement, we find that the sensor has a resonance frequency of 14.7 Hz in its x direction, with a Q factor of 15.3. In the y direction, we find a resonance frequency of 14.0 Hz and a slightly lower Q factor of 15.1. This slight difference is most visible near the phase flip and amplitude maxima and is reflected in their different fitted values for N_r. This difference can therefore be attributed to the mechanical properties of the non-perfect square lattice inside the fiber support structure; a slight directional difference exists.

Figure 2.7 shows the wavelength shift signal and body motion over time of five trials of the frequency response measurement. As expected, the relative orientation and magnitude of both types of data match up. From the ratio of sensor signal amplitude to sensor motion amplitude, we obtain a measured geometric factor of $3.58 \cdot 10^{-7}$ in the millimeter domain under dynamic conditions.

Due to the small directional difference in bending stiffness and resulting frequency response, the x and y projections of motion, and therefore sensor signal, have a slight phase lag difference. This produces sensor motion in ellipses rather than straight lines at stimulation close to either x or y resonance frequencies. In the next section we demonstrate a method to correct for this phenomenon.

2.5 DISCUSSION

The measured sensor behavior has been shown to compare well with the modeled input-output and frequency characteristics. The sensor acts as a dampened resonator both driven and dampened by fluid forces. The results show that the measured frequency response (figure 2.6) is accurately described by the combined hydrodynamics and strain-structure model. Both large, static, mechanical deflection ($G = 3.56 \cdot 10^{-7}$) and small, dynamic, hydrodynamic stimulation ($G = 3.58 \cdot 10^{-7}$) aligns with the modeled mechanical properties of the sensor based on the cross section from the sensor tip ($G = 3.22 \pm 0.05 \cdot 10^{-7}$).

This model can therefore be used to relate sensor signal via sensor body motion to local fluid flow in two steps. First, the sensor signal can be translated to sensor body motion with the geometric factor G. Then, by a deconvolution via the frequency response (eq. 2.14), which incorporates the fluid-dynamic characteristics of the sensor, we can accurately obtain the local 2D flow velocity (see figure 2.8).

2.5.1 *Sensitivity and dynamic range*

The measurement of the frequency characteristics in figure 2.6 shows an expected peak, which indicates the resonance frequency, where the sensor is most sensitive. Higher frequencies will be filtered out, i.e. there's a constant downward slope in the frequency response curve. At frequencies below resonance, the frequency response plateaus and will still pick up low frequency and DC flow (see eq. 2.20 and table 2.1), although it is not specifically designed to do so.

The lowest detectable flow speeds of the artificial neuromasts affect the detection limits for tracking objects using artificial lateral lines [20]. We can infer this threshold velocity, at which the expected sensor signal equals noise levels, by taking the ratio of the measured noise in $\boldsymbol{\delta\lambda}$ to G times the frequency response (table 2.1).

When down sampling the signal by averaging with a factor 50, the noise levels dropped from 2.02 pm to 0.28 pm with a factor close to $\sqrt{50}$, as would be expected from normally

Table 2.1: Lowest detectable fluid velocities for the optical sensor. The lower frequency sensitivities are extrapolated from the fitted model and indicated with a $*$. Both x, y resonance frequencies are displayed to indicate the magnitude of difference between sensing low frequency flow and flow at sensor resonance.

	SAMPLING RATE & DETECTABLE VELOCITY			
	5 kHz	5 kHz	100 Hz	100 Hz
STIMULUS	V_x (mm/s)	V_y (mm/s)	V_x (mm/s)	V_y (mm/s)
0 Hz*	36	34	5.1	4.8
0.01 Hz*	22	21	3.2	3.0
0.1 Hz*	9.1	8.5	1.3	1.2
1 Hz*	1.7	1.6	0.23	0.22
10 Hz	0.11	0.098	0.016	0.014
14.0 Hz^y	0.031	0.027	0.0043	0.0039
14.7 Hz^x	0.028	0.030	0.0039	0.0043
50 Hz	0.42	0.44	0.059	0.062

distributed noise. The sensor is shown to be linear up to deflections of about 5 mm (figure 2.5). Taking the corresponding 1.78 nm wavelength shift and σ_{noise} as the upper and lower boundaries respectively, this amounts to a linear Dynamic Range of 29 dB at 5 kHz sampling and 38 dB when down sampled to 100 Hz.

2.5.2 *Two-dimensional fluid flow sensing*

Two orthogonal projections of flow can be measured by the all-optical artificial neuromast using the information of at least three cores. This requires that four fibers are positioned in a square lattice embedded in a support structure. The sensor cross section as shown in figure 2.3b shows that the centers of the four fibers do not form a perfect square, so some directional variance in stiffness and therefore sensitivity exists. By employing a 2D method of sensing sensor body motion, we can measure, and correct for, this directional variance. In practice, by processing the sensor signals via a deconvolution along two orthogonal dimensions, we can translate sensor signal and body motion with their respective phase lag and resonance properties to a 2D representation of local fluid flow.

Figure 2.8 shows the deconvolution process (for details, see chapter 17 of Smith's book [108]). In this process, measured sensor signals are first band pass filtered in order to reduce high and low frequency noise. In the frequency domain, the real sensor amplitudes are divided by the amplitudes of the FR. The FR phase is subtracted from

the imaginary valued sensor phase information. Using an inverse Fourier transform, the resulting amplitude and phase information is then transformed back into the time domain. For longer or continuous sensor readings, taking a small buffer period into account, this procedure can be performed real-time using sliding time windows.

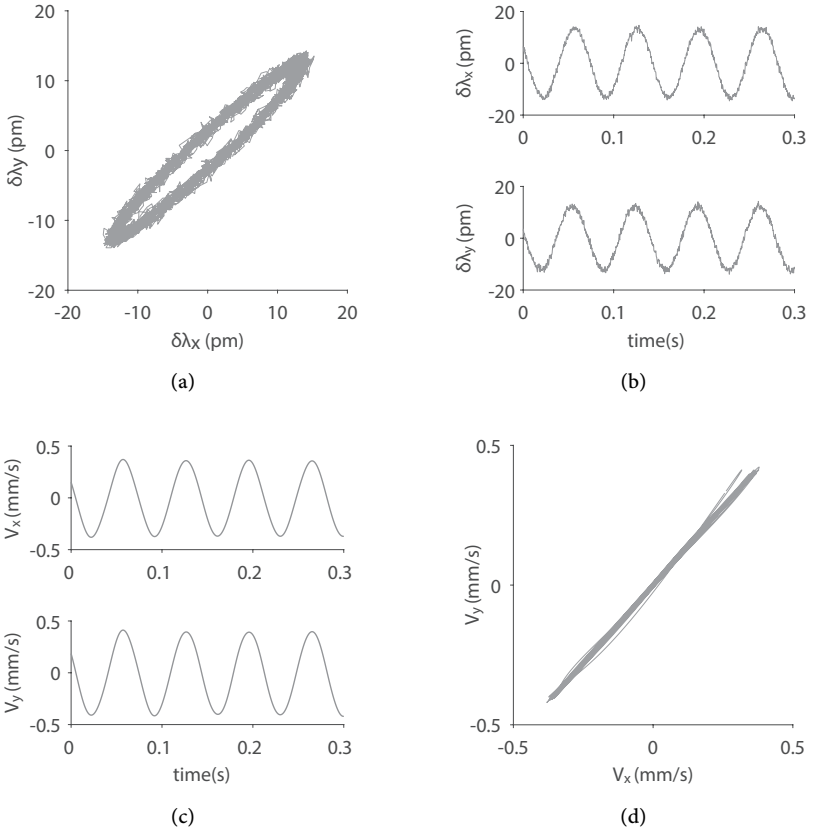


Figure 2.8: Progression from raw sensor signals at 14.4 Hz stimulation to local velocity reconstruction. (a) The 2D representation of sensor signals shows an elliptical motion due to the phase difference in the x, y projection. (b) Section of x, y signal. (c) After band-pass filtering the signal, applying the geometrical factor and deconvolving via the FR, the slight phase difference between x and y is reduced. (d) 2D projection of reconstructed velocity.

In addition, we can express the uncertainty of the sensor readings both in Cartesian projections of measured fluid flow as well as a magnitude and direction representation. As is clear from figure 2.8d, the reconstructed velocity contains some magnitude and direction variance. Ignoring velocity readings under 0.05 mm/s and taking the ratio of

V_x and V_y over time, the variation in the reconstructed direction of velocity oscillation amounts to a standard deviation of $\sigma = 1.8°$, with an oscillating flow amplitude of 0.54 mm/s. The sensor is therefore able to reliably determine fluid flow direction within a few degrees.

2.6 CONCLUSION

We have shown that the fiber structure mechanically acts in accordance with our combined hydrodynamics and strain-structure model. The predictable nature of dynamic behavior allows for deconvolution of measured strain induced FBG signals via the frequency dependent velocity sensitivity which, in turn, allows for direct translation to fluid velocity. The dynamic characteristics are tunable by varying the sensor dimensions and mechanical properties, allowing a tailored design for specific use cases.

The presented strain-structure model allows optimizing the sensor design for generating strain and thus sensor signal per sensor motion. The observed geometric factor relating sensor body motion to observed differential wavelength shift is in accordance with the modeled geometric factor. It is consistent throughout the measurements and allows individual sensors to be calibrated by using conventional displacement sensitive methods, such as mechanical deflection or visual monitoring.

A welcome enhancement from its 1D-sensitive biological counterpart and state-of-the-art artificial neuromasts is that by interrogating the sensor in two perpendicular axes, a single sensor will provide information about both the magnitude and the angle of the local flow velocity. Although it is shown that stationary artificial lateral lines can reconstruct dipole sources and moving artificial lateral lines can detect obstacles without a second dimension, this extra information might be able to increase the ALL detection precision and effectiveness.

Part II

SIMULATION STUDIES

This part describes a series of simulation studies in three chapters, which are used as a basis for experiments performed in part III.

Chapter 3 compares the performance of several artificial neural networks for localizing an object in a plane using an array of 1D-sensitive sensors. Using a heat map approach in chapter 4, the system is extended to perform multi-object localization in a 3D volume. In Chapter 5, arrays of 1D-sensitive and 2D-sensitive sensors are compared with respect to single object localization to demonstrate the advantage of 2D-sensing.

3

SIMULATED 1D-SENSITIVE LOCALIZATION

In this chapter, we compare three types of regression neural networks for localizing a moving object. The multi-layer perceptron (MLP), the echo state neural network (ESN), and the extreme learning machine (ELM) are compared with respect to their ability to reproduce the coordinates of the moving object, based on a sampled velocity pattern as input.

The simulation consists of a sphere moving in a plane with constant speed, using a simplified hydrodynamic model. This sphere moves in a bounded rectangular area, changing its direction to produce random curved paths of motion.

We consider both the localization error distance (MED), and the execution time as important measurements for determining what type of neural network architecture is better for moving object localization. In addition, we study the influence and sensitivity of these trained networks to noise by varying the signal to noise levels.

PROVENANCE This chapter has been previously published as:
L.H. Boulogne, B.J. Wolf, M.A. Wiering, S.M. van Netten, (2017). Performance of neural networks for localizing moving objects with an artificial lateral line. *Bioinspiration & Biomimetics* 12(5) 056009.

AUTHOR CONTRIBUTIONS LB implemented the initial simulation and neural networks, and co-wrote the draft. BW performed the final ELM and ESN validation, performed the spatial analysis, co-wrote the draft, and wrote the final manuscript. MW co-supervised the study and revised the draft. SvN conceived the study, supervised the study, and revised the draft.

ABSTRACT Fish are able to sense water flow velocities relative to their body with their mechanoreceptive lateral line organ. This organ consists of an array of flow detectors distributed along the fish body. Using the excitation of these individual detectors, fish can determine the location of nearby moving objects. Inspired by this sensory modality, it is shown here how neural networks can be used to extract an object's location from simulated excitation patterns, as can be measured along arrays of stationary artificial flow velocity sensors. The applicability, performance and robustness with respect to input noise of different neural network architectures are compared. When trained and tested under high signal to noise conditions (46 dB), the Extreme Learning Machine architecture performs best with a mean Euclidean error of 0.4% of the maximum depth of the field D, which is taken half the length of the sensor array. Under lower signal to noise conditions Echo State Networks, having recurrent connections, enhance the performance while the Multi-Layer Perceptron is shown to be the most noise robust architecture. Neural network performance decreased when the source moves close to the sensor array or to the sides of the array. For all considered architectures, increasing the number of detectors per array increased localization performance and robustness.

3.1 INTRODUCTION

Along the sides of their body, fish have a mechanoreceptive lateral line organ that enables them to detect nearby moving underwater objects producing local water flow [27, 36].

This sensory lateral line organ consists of an array of individual detectors, called neuromasts, which are sensitive to water flow velocity. Each neuromast contains hair cells that detect the movement of the water at the location of that neuromast [41]. The lateral line organ is used for a variety of different tasks. It allows fish, also when no light is available, to detect e.g. prey [61] and predators and facilitates schooling [98]. The related sensory modality is sometimes described as in between touch and hearing and is sensitive to the near field component of pressure gradients [68, 92].

The present work is inspired by the lateral line organ and is intended to be used in the signal processing and interpretation of excitation profiles measured along artificial arrays of individual flow velocity sensors to efficiently localize moving objects.

In previous research, dipole fluid flow models are used which predict excitation patterns along a stationary array, given a source that is vibrating in a direction with a specific angle with respect to the array [6, 31, 33, 50] or moving in a specific direction [43], all under conditions of potential flow.

Using these models, excitation patterns for different locations and directions of a moving spherical source can be generated. Several neural network architectures are considered in the present work for their ability to accurately decode the location x, y of a moving object from the excitation patterns along a stationary artificial sensor array.

For both simulated and physical artificial lateral line arrays, several algorithms have been put forth to decode a dipole-like source location from excitation patterns. A data-

matching approach where a measured excitation pattern is compared to a large set of templates was used by Pandya et al. [96]. In later studies [125] these excitation patterns were matched to a gaussian which is a crude approximation of its wavelet nature [31]. This matched (gaussian) filter approach was later shown to be outperformed by Capon's beamforming algorithm [127].

In [4], Abdulsadda and Tan showed a relatively small Multi-Layer Perceptron (MLP) neural network with maximally 24 hidden nodes and 6 input nodes that was able to decode a dipole source location. With the length of the array as BL (body length) and in a $2BL \times BL$ area, their reconstructed location has an average Euclidian error of 1.5% BL on a sparse (<100 samples) data set. From the supplied typical sensor signals Abdulsadda and Tan [4, p. 234], we estimate a signal to noise ratio of about 30 dB. In later research, the authors remark that "due to the black-box nature, that approach requires a lot of training data unless the dipole vibration amplitude and orientation are known" [5], but concerns a limited parameter set and optimization scheme.

The present work focuses on processing lateral line excitation patterns using neural networks. By exploring and optimizing several neural network architectures for a generic stationary velocity sensor array, the results may therefore be considered to be independent from particular stationary sensor array characteristics.

An attractive property of neural networks when used in combination with operational velocity sensing arrays, is that the neural network also can take into account and correct for variations in the individual physical sensors characteristics and noise. Furthermore, these biomimetic signal processing methods also allows for rectified parallel processing as observed in fish [21].

The network types used in this research are Multi-Layer Perceptrons (MLP) trained with the back-propagation algorithm [103], Extreme Learning Machines (ELM) [64] and Echo State Networks (ESN) [66].

We have selected these three different neural network methods, because they are well established and have different advantages and disadvantages, which allows for interesting comparisons. Multi-Layer Perceptrons have the advantage that the features which the hidden units extract from the inputs are learned by using the back-propagation algorithm. This makes them slower than Extreme Learning Machines, which initialize the input to hidden unit weights to random values which are then not trained further. The Echo State Network was chosen, because it has the property that it can use previous inputs from the time-series signals and therefore it can use more information. This is at the cost of being governed by more complex dynamics than the other models. With these choices, we will have the opportunity to observe which method performs best whether partial training or memory affects localization performance.

3.2 METHODS

3.2.1 *Data generation*

The MLPs, ELMs and ESNs are trained using a training set and their performance assessed by a test set. With the MLPs, a third set of data (validation set) is also used to avoid overfitting during training. For these three sets, three different trajectories of source object movement are used, see section 3.2.1.2.

3.2.1.1 *Computing water velocities*

The data sets used in this research resemble the information perceived by the artificial lateral line organ and consist of simulated hydrodynamic data. For the construction of the data sets the fluid velocities caused by a source object (sphere) moving in a 2D plane in a 3D volume through water are calculated. Figure 3.1 shows this scheme.

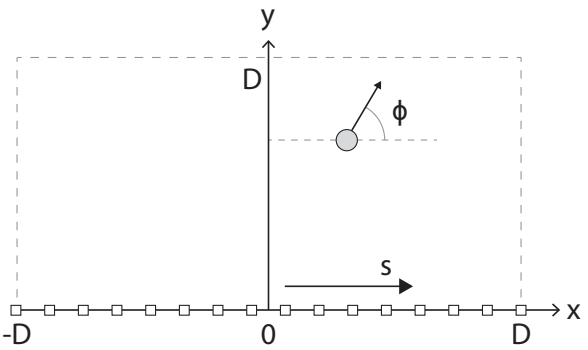


Figure 3.1: Top view of geometry. The spherical source moves with a fixed velocity in the x,y-plane in a direction that has a variable angle ϕ with the direction of the stationary artificial lateral line array. The position of a sensor on the simulated array is denoted by s.

The sensor array is located along the trajectory running from coordinate $(-D,0)$ to $(D,0)$. The sensors are equally spaced along this line. The first sensor is located at coordinate $(-D,0)$ and the last sensor at $(D,0)$.

An experimental justification for using potential flow in a similar set up as assumed in the present study has been obtained in studies on fish. Both the shape of the wavelets and specifically the distance coding in its spatial characteristics predicted by potential flow have been observed in the biological lateral line responses of fish [31]. It was shown that the boundary layer of the stationary fish was not affecting the flow field to a high extent, which can also be expected using artificial lateral line sensors. This observation is supported by a theoretical study by Goulet et al. [50] and by a computational fluid dynamics study [100].

In Franosch et al. [43] a similar method to find the fluid velocity distribution for a moving source object (sphere) with a constant speed is considered. This method too was used for the case of inviscid potential flow. The velocity field distribution described in that work can be shown to be equal with the distribution in equation 3.1. This implies that the present work can equally well be applied to a vibrating source at different locations as well as moving sources.

The fluid velocity component parallel to the array at position s on the sensor array, $v(s)$, is given by

$$v(s) = C(\psi_o \sin\phi - \psi_e \cos\phi), \tag{3.1}$$

with

$$C = \frac{Wa^3}{2y^3}, \tag{3.2}$$

where W is the velocity of the source object and a is the radius of the sphere. The angle of the source with respect to the sensor array in radians (see figure 3.1) is ϕ, and the even wavelet ψ_e and the odd wavelet ψ_o are described respectively by

$$\psi_e(s,x,y) = \frac{1 - 2\left(\frac{s-x}{y}\right)^2}{\left[1 + \left(\frac{s-x}{y}\right)^2\right]^{\frac{5}{2}}}, \tag{3.3}$$

$$\psi_o(s,x,y) = \frac{-3\left(\frac{s-x}{y}\right)}{\left[1 + \left(\frac{s-x}{y}\right)^2\right]^{\frac{5}{2}}}. \tag{3.4}$$

Here (x,y) denotes the instantaneous coordinate of the moving object. The equations 3.3 and 3.4 show that the shape of the even and odd wavelets solely depends on the location of the source object with respect to the sensor location. The shape of these wavelets is shown in figure 3.2.

In Curcic-Blake and Netten [31, pp. 1551] it was noted that the spatial variations along the x direction, as described by the even and odd wavelet functions, scale linearly with the distance of the source y. It was also shown that the maximum amplitude of the even wavelet is reached at the point of the lateral line that is closest to the source and that the odd wavelet is zero at this position. This is the location on the lateral line that is equal to the x coordinate of the source object.

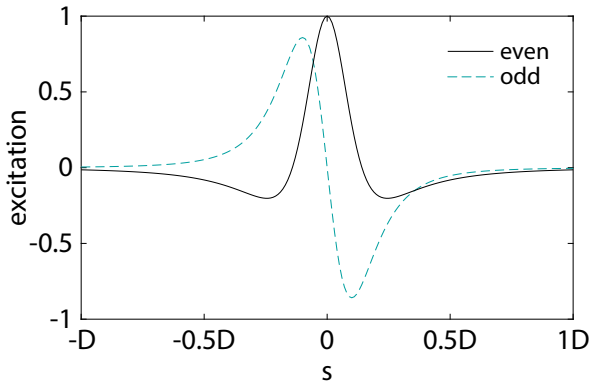


Figure 3.2: The even and odd wavelets ψ_e and ψ_o for a moving source located at coordinate (0, $0.2D$). The wavelets are defined in equations 3.3 and 3.4 respectively and scale spatially with the distance to the array y.

3.2.1.2 *Paths of sphere movement in the data set*

Given the implicit memory present in the ESN architecture, the hydrodynamic data for all architectures is presented in trajectories rather than discrete locations so that for predicting the current location, the internal representation of past detections can be used.

The source object starts at time $t = 0$ at a random location (x_0, y_0) in the Cartesian system where x_0 is taken from a uniform distribution with range $[-D, D]$ and y_0 is taken from the uniform distribution with range $[0, D]$. The object remains located in a $2D \times D$ area to one side of the lateral line (see figure 3.1).

For the data sets it is assumed that the source object moves with a constant velocity of $0.1D$ per time step. The direction ϕ_t in radians in which the object moves at time step t is

$$\phi_{t+1} = \phi_t + A. \tag{3.5}$$

Here, A, in radians, is taken from the uniform distribution with range $[-1, 1]$. The change in angle per time step is therefore limited to about 60 degrees.

The next location at time $t+1$ is selected by moving the sphere in the direction denoted by angle ϕ_t over a distance $0.1D$.

When the source object would move outside of the area boundaries, the next movement direction is altered as shown in figure 3.3. If the object would still cross a boundary when it is moved in the resulting direction, the movement direction is similarly changed again.

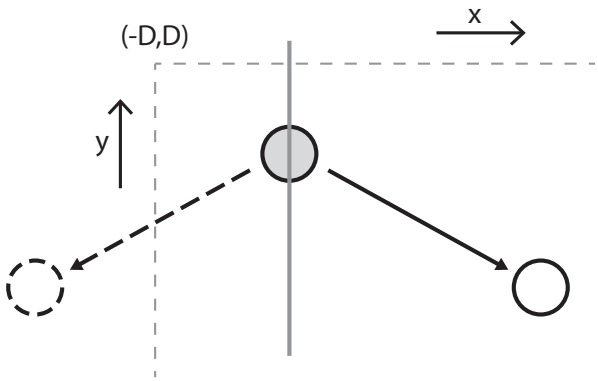


Figure 3.3: The source object at time t (gray sphere) would cross the field boundaries if it would be moved in the original movement direction along the dashed movement vector. Therefore, instead it is moved in a different direction along the different solid movement vector. This new (solid) movement vector is obtained by mirroring the original (dashed) movement vector, where the mirror (grey vertical) line is centered on the source object and parallel to the field boundary that is about to be crossed. The solid white sphere shows the location of the sphere at time $t + 1$.

3.2.1.3 *Splitting water velocities into rectified inputs*

Each data point in the data sets represents the fluid velocities detected by sensors that are equally spaced along a one-dimensional sensor array. This concatenation of fluid velocities is used as input for the neural networks.

In fish neuromasts, two types of velocity detecting hair cells are present. The first type only detects water flow in one direction, while the second type only detects flow in the opposite direction [41]. The two sorts of information perceived by the two types might be transmitted separately to the central nervous system [90, pp. 290]. If this is so, the fish is able to discern between positive and negative fluid velocities.

As a preprocessing step, this biological parallelization of information is imitated; each sensor reading is represented using two values. The first value only represents positive velocities. It is zero for negative velocities. For the second value, the same applies, but vice versa. Since lateral lines with 16 and 32 neuromasts are simulated, this results in using 32 and 64 inputs. We used this input doubling because of the observed enhanced performance in source localization.

3.2.1.4 *Adding noise*

To test the noise robustness of the networks, different levels of noise are added to the fluid velocities computed for test sets.

The noise is taken from a normal distribution with a mean of zero. Different noise levels are applied via variation of the standard deviation of the noise. The noise level is defined in this chapter in terms of the Signal Noise Ratio (SNR), which is given as:

$$\mathrm{SNR} = 10 \log_{10} \frac{A^2}{\sigma^2} dB \tag{3.6}$$

In this equation, A is the maximal magnitude of deviation from zero of an input excitation pattern after scaling (section 3.2.1.5) has been applied. σ^2 is the variance of the normal distribution out of which the noise values are taken.

The distance from the source to the artificial lateral line y affects the amplitude of the signal in the excitation pattern (see equation 3.2). To investigate the effects of noise independent of y, noise is added to all sensor signal inputs to obtain the required signal to noise ratio and then scaled, according to equation 3.7.

Adding noise affects the range of the values in an input excitation pattern. In real-life applications in which noise quantities are unknown, all excitation patterns would be scaled to the same value ranges, regardless of the quantity of noise that is present. As a consequence, the contribution and level of the original signal to the normalized excitation pattern differs for each noise level.

3.2.1.5 *Scaling the input*

In order to make the localization process robust to amplitude variance and to let it focus on the spatial characteristics of the excitation pattern, it is first scaled before the input is presented so that the largest magnitude in the excitation pattern becomes 1. For this, all excitation pattern values are changed according to equation 3.7.

$$\text{new } q[n] = \frac{q[n]}{\max |q|} \tag{3.7}$$

The discrete signal q is the excitation pattern at each individual sensor calculated with equation 3.1 and index n is the sensor number. This scaling causes the values of q to always be within the range [-1,1]. This causes information about the y coordinate to be present only in the spatial scaling of the normalized excitation pattern, while information about the x coordinate is present in the location of the pattern along the sensor array.

3.2.2 *Neural network algorithms*

3.2.2.1 *Multi-layer perceptron*

The MLP used in this study is a fully connected network (see figure 3.4). The input layer has a variable size (32 or 64 nodes) and the output layer has two nodes to represent the

location x, y of the source. Equation 3.8 shows the way in which the node activation values are computed.

$$x_i^{m+1} = f\left(\sum_{j=1}^{N^m}\left(w_{ij}^m x_j^m\right) + b_i^{m+1}\right) \tag{3.8}$$

Here, x_i^m denotes the activation of the ith node of the mth layer, where counting starts at the input layer. w_{ij}^m denotes the weight connecting the jth node in the mth layer to the ith node in layer $m+1$. b_i^m denotes the bias of the ith node in the mth layer. N^m is the number of nodes in the mth layer. The activation function $f(x) = tanh(x)$ is used to calculate the activation of the hidden nodes. For the output nodes, the identity function is used as activation function (see figure 3.4). During training, when computing the activation value of a node in the hidden layer, some noise sampled from a uniform distribution with range [-10^{-3}, 10^{-3}] is added, which amounts to 46 dB.

The network was trained with the incremental learning version of the backpropagation algorithm [103]. The weights are initialized according to the normalized initialization procedure described by Glorot and Bengio [47]. The bias weights are initialized to zero.

When overfitting occurs, a network is trained to specifics of a training set instead of general features. This causes the network to perform better on the specific training set, but worse in general cases.

To determine whether overfitting occurs, during training, the Mean Squared Error (MSE) on the training and a validation set was monitored. This is the usual error that is minimized during training neural networks. The MSE for a network on a specific data set is

$$\text{MSE} = \frac{1}{D^2 M}\sum_{n=1}^{M}\frac{1}{N}\sum_{i=1}^{N}(t_i(n) - o_i(n))^2. \tag{3.9}$$

In this equation D is half the length of the array, M is the number of samples in the data set, $N = 2$ is the dimensionality of an output sample, t is the target output and o is the network output.

An additional measure for reporting is the Mean Euclidean Distance (MED):

$$\text{MED} = \frac{1}{DM}\sum_{n=1}^{M}\sqrt{\sum_{i=1}^{N}(t_i(n) - o_i(n))^2}. \tag{3.10}$$

The MED provides a more intuitive relative distance measure of error as it is defined as a fraction of the depth of field D.

3.2.2.2 *Extreme learning machine*

Like an MLP, an ELM is a feedforward neural network (see figure 3.4). It differs from the MLP in that only the weights from the hidden to output layer are trained. Weights from

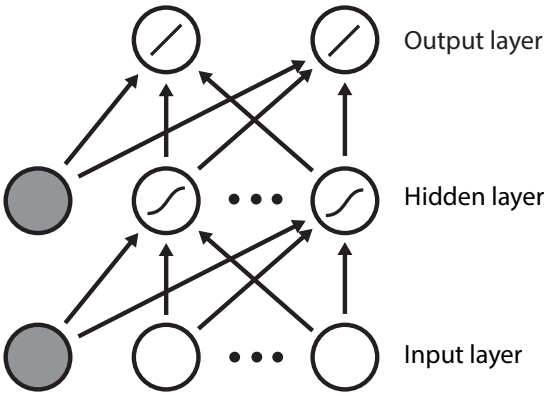


Figure 3.4: Visual representation of the feed forward networks used in this research (MLP and ELM). Weights are represented by arrows. The activation functions are indicated in the nodes. The grey nodes represent bias nodes with a fixed value of 1.

the input to the hidden layer are initialized randomly and not altered during training [64].

An ELM with a perfect performance on the training set would have a weight matrix W, that describes the weights from the hidden layer towards the output layer, that satisfies:

$$WH = T. \tag{3.11}$$

Here, the teacher output T is the matrix that is built up from a consecutive series of columns, in which each column consists of the correct output for all output nodes to the training pattern. Hidden layer output matrix H is built up from the activation values of the units in the hidden layer at all the time instances during the training that are the result of presenting the training data to the ELM.

Training of the ELM consists of finding a least squares solution for W from this linear system, given H and T. This is computed using:

$$W = TH^{\dagger}, \tag{3.12}$$

where $H^{\dagger}$ is the Moore–Penrose generalized inverse of matrix H.

Because determining the matrix W is very fast, ELMs can be efficiently trained in little time compared to MLPs [64].

3.2.2.3 *Echo state networks*

While experimentally determining parameter settings, ELMs with internal recurrent connections were also tested. The resulting ESN [66] architecture effectively introduces memory into the neural network, which could help in predicting the current position of a source based on past detections.

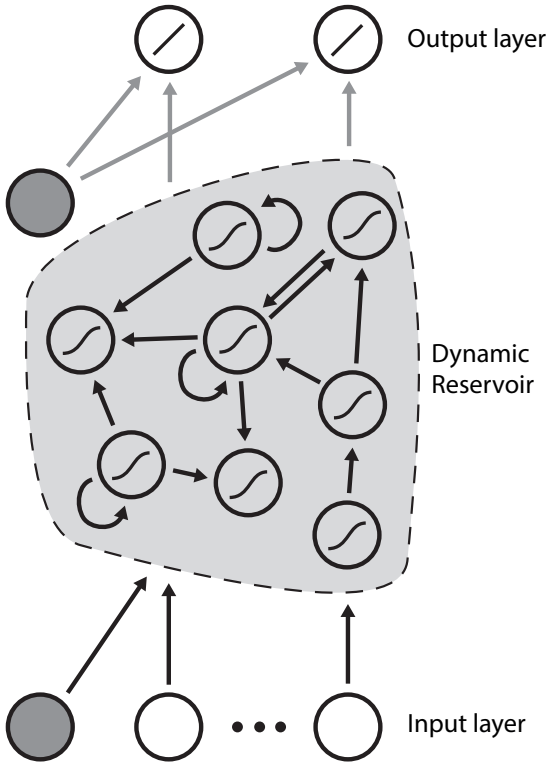


Figure 3.5: Schematic representation of a basic ESN. Weights are represented by arrows, grey nodes represent bias nodes. Only the (grey) weights from the dynamic reservoir to the output layer are trained, black weights are selected from a uniform distribution. After [66, p. 7]. Without recurrent connections the Dynamic Reservoir collapses into a single hidden layer (like an ELM).

Figure 3.5 shows the schematic representation of an ESN that consists of an input layer, a dynamic reservoir (DR) and an output layer. Each input node connects to each node in the DR and each node in the DR connects to the output layer through weights. The hidden nodes in the DR are sparsely connected with each other and themselves. This causes the DR to contain a high dimensional representation of the input.

Table 3.1: Computation times for network architectures on a 2.5 GHz Intel i5 core processor. The train and test sets both contain 15000 samples (excitation profiles). The average computation time per single sample for a trained network is also listed.

	COMPUTATION TIME		
NETWORK	TRAINING	TEST SET	SAMPLE
MLP_{16}	9 hr	4.6 s	0.3 ms
MLP_{32}	9 hr	4.7 s	0.3 ms
ELM_{16}	100 s	2 s	0.1 ms
ELM_{32}	240 s	3 s	0.1 ms
ESN_{16}	160 s	48 s	3.2 ms
ESN_{32}	382 s	100 s	6.6 ms

The weight matrix representing the internal weights of the DR must be constructed so that the spectral radius, which is the largest absolute eigenvalue of the weight matrix, is smaller than 1. As a result input is echoed and dies out over time, which is called the Echo State Property. This gives the ESN short term memory.

Due to the short term memory, the output of the first few steps is often inaccurate. To wash out the effects of the initial network state, the network output of the first 50 samples is discarded during training and testing.

3.3 RESULTS

To optimize the meta-parameters of the neural network models, we performed many trials with different values and selected the best parameters for the final experiments. After determining the optimal meta-parameter settings of the networks, new data sets from new trajectories were generated to train and test the networks that produced the results in section 3.3. The training and test sets used each contain 15000 samples. A large number of samples was chosen to make sure that enough source locations were well represented. The validation set contains 10000 samples.

Table 3.1 indicates the time required for training the neural network; the time it takes for the test set to be parsed and finally the calculation time for a single excitation pattern on a 2.5 GHz Intel i5 core processor.

3.3.1 *Parameter settings*

Below, we first show the performance of networks with the best parameter settings found in the parameter study for high signal to noise data (46dB).

Table 3.2: Best performing localization results for the MLP and ELM and their respective MEDs.

NETWORK	MSE	MED
MLP_{32}	5.23×10^{-5}	0.68%D
ELM_{32}	3.66×10^{-5}	0.41%D

3.3.1.1 *MLPs*

For optimizing the MLPs, the varied parameters were the learning rate, number of hidden layers and layer sizes. All MLPs in the initial parameter study were trained for 300 epochs. The best performing networks for both 16 and 32 sensors were then trained with a learning rate of 0.01. They had two hidden layers of 80 and 40 nodes respectively. For these optimal parameter settings, new networks were trained for 3000 epochs.

3.3.1.2 *ELMs and ESNs*

The best performing and thus chosen hidden layer sizes were 5000 and 7000 nodes for ELMs with 16 and 32 sensors respectively. The performance of the ESN architecture increased when the recurrent connection weights were chosen smaller. Since the performance was best in a neural network architecture without recurrent connections, the best performing ESNs are effectively ELMs, and only ELMs are discussed for high signal to noise input.

In order to ascertain and compare the noise robustness of the ESN architecture, ESNs with DR sizes of 5000 for 16 inputs and 7000 nodes for 32 inputs were also tested with a spectral radius of 0.1.

3.3.2 *Overall performance on high signal to noise input*

To investigate the differences in performance between the different network architectures and different array sizes, 5 ELMs and 5 MLPs for both 16 and 32 input sizes, indicated with a subscript, with the best performing parameter settings were trained and tested with new data sets.

The random initialization of the weights of the networks resulted in different networks per combination of network type and input size. Using multiple networks per combination of input size and network type, maps this variety to the output error per individually generated and tested network.

The boxplots in figure 3.6 show the MSE distribution per network type on respectively the x and y coordinates as well as the average MSE. The network instances with the lowest average error are listed in table 3.2. For comparison, when forcing a spectral

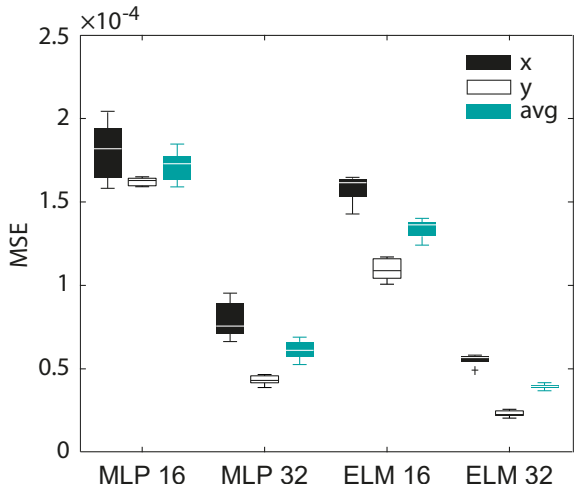


Figure 3.6: Boxplots of the MSE and specified for the x and y coordinate per network type.

radius of 0.1 on an ESN, the MED localization error compared to the ELM approximately doubled to 0.71%D.

Figure 3.7 gives an indication of source localization performance of the best performing ELM and MLP under high signal to noise conditions.

In figure 3.8, the effect of the source location on the MED performance is separately shown for both the x and y coordinate.

3.3.3 *MLP overfitting*

For all MLP_{16} and MLP_{32} networks, the MSEs on the training set were compared with the MSEs on the validation set for every epoch during training. An example of such a comparison can be seen in figure 3.9. For no MLP trained and tested with the new data set, the MSE of the validation set increased continuously, or to a stable value at any point in training. Therefore, we conclude that overfitting does not occur in these networks.

3.3.4 *Noise robustness*

The MSE of all three network architectures was found for different SNRs. The SNRs were chosen to lie in the interval [-6 dB, 46 dB]. The results are plotted in figure 3.10a.

Figures 3.10b and 3.10c show the noise robustness separately for respectively the x and y coordinates of the source object.

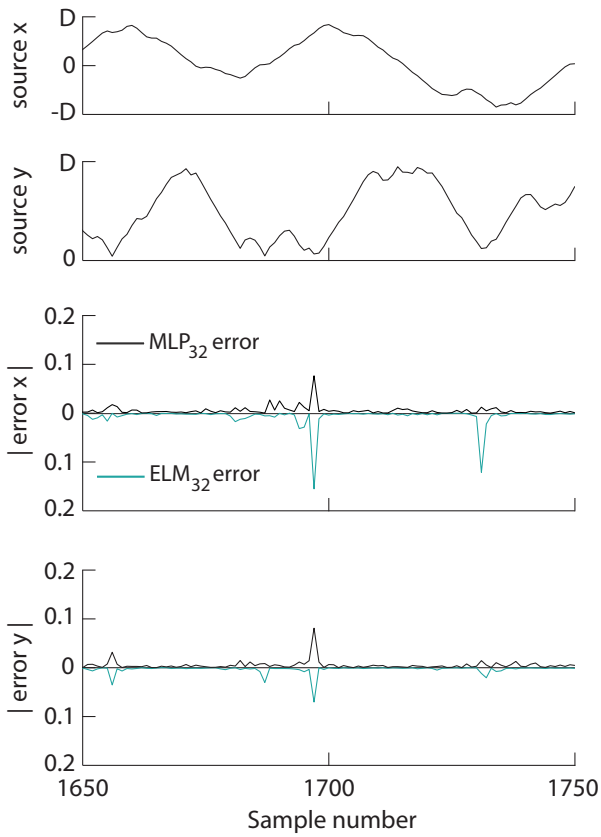
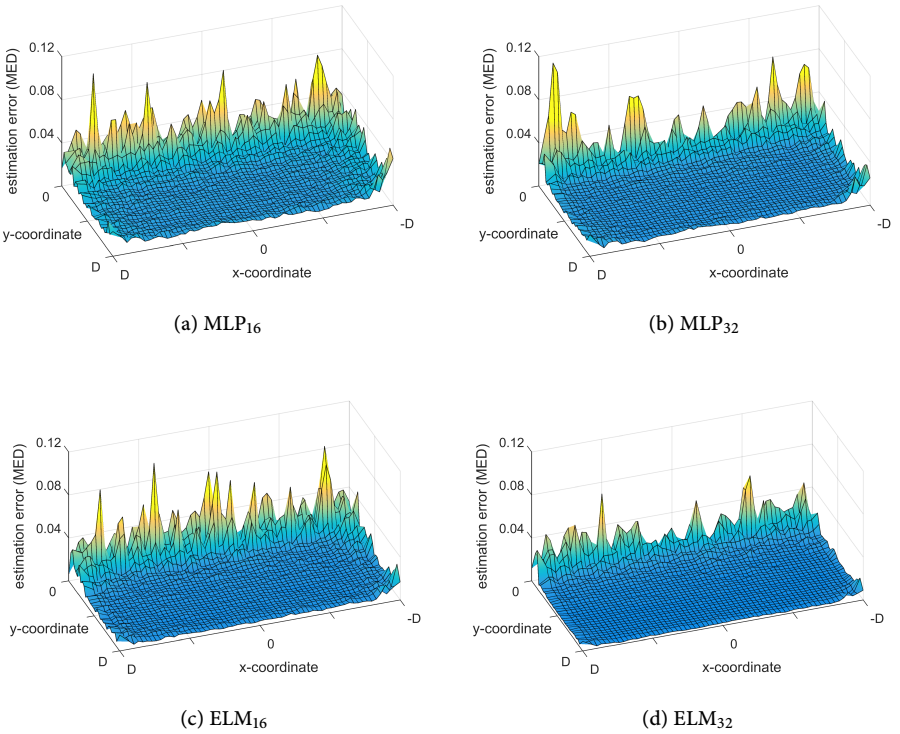


Figure 3.7: Actual source location and absolute Euclidean error (as a fraction of D) on a part of the test set of individual networks of the ELM_{32} and MLP_{32} sets for x and y coordinate. To enable a clear visual comparison of the neural network errors, the absolute errors of the MLP and ELM are shown in mirrored y-axes. The peak in error around sample 1700 coincides with the source being located in a corner. The MLP is less affected by this corner effect than its ELM counterpart. The ELM error is however generally lower.



(a) MLP_{16} (b) MLP_{32}

(c) ELM_{16} (d) ELM_{32}

Figure 3.8: Interpolated performance (MED) for the considered networks based on the source position x, y. The same test set is used for each network type. The estimation error increases around the edge of the detection field and shows an increased error close to the array, i.e. when y is close to zero. This may result from an effective undersampling of the excitation pattern, where the spatial characteristics such as extrema and zero-crossings (see figure 3.2) cannot be readily inferred since the pattern is spatially narrowed beyond the inter-sensor distance.

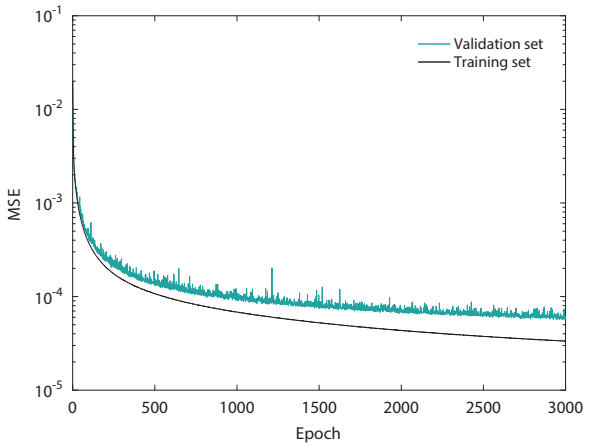


Figure 3.9: An example of the MSE on the validation set and on the training set for the best performing MLP_{32}.

For comparison, figure 3.10d shows the noise robustness of ELMs and ESNs with a smaller hidden layer size than optimal (see section 4.2).

3.4 DISCUSSION

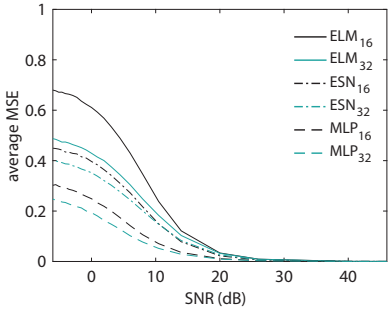
3.4.1 *Practical implementation neural networks*

From the results shown in figure 3.7 it can be concluded that neural networks are able to reliably extract information about the location of a moving source in a field with a depth D equal to half the sensor array length with an accuracy of the order of percents of D, given the simulated dipole fields.

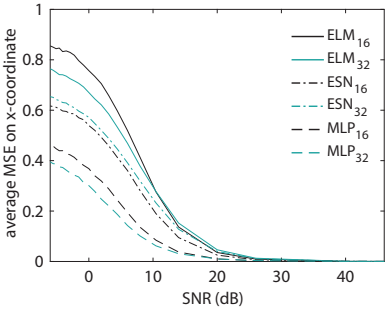
On average, network architectures with 32 sensors outperformed those with 16 sensors, as seen in figure 3.6. The limited variance of performance within the network types suggests that the networks don't suffer from local minima.

Depending on the computing power and response requirements, trained networks may be used under real time conditions. On our system, in combination with an ELM, MLP, or ESN we have typical response times of 0.2, 0.3, or 6 ms respectively for updating a source location.

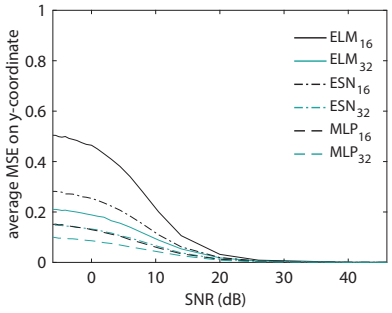
The consequence of having a finite response time is that estimates on source location can only be generated with a delay t_{update}, which results in the source having moved a distance Δd, equal to $\Delta d = V \cdot t_{\text{update}}$, with respect to the position it was during the array velocity measurement. It is useful to compare this distance Δd, with the spatial



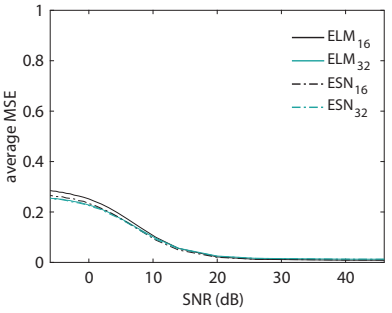
(a) Noise robustness



(b) Noise robustness for x



(c) Noise robustness for y



(d) Reduced network size

Figure 3.10: This figure shows the performance (in MSE) of the different neural network architectures. Here (a) shows the average performance of each neural network type, which is portrayed separately in (b) and (c) for the x- and y-coordinate respectively. (d) shows the noise robustness of ELMs and ESNs with a limited number (100) of hidden nodes.

accuracy of the source's location. If we accept a similar inaccuracy of distance Δd, as the inaccuracy produced by the neural network, given by the MED (i.e. $\Delta d = \text{MED} \cdot D$) we arrive at an upper bound of velocities ($V_{\max}$) which may reliably be detected:

$$V_{max} = \frac{\text{MED} \cdot D}{t_{\text{update}}}. \tag{3.13}$$

Clearly this velocity is proportional to array length ($2D$) and inversely proportional to the update time.

For instance, with a MED of 0.05 (at least feasible with an ELM and a SNR velocity input of 22 dB and considering $t_{\text{update}} = 0.2$ ms per sample with our computer platform) we arrive at an upper bound for the velocity equal to $V_{\max} = 250$ D/s. This example

indicates that a neural network under realistic conditions will allow the processing of source velocities two orders of magnitude higher than the array length per second, which entails that most likely other factors than the update speed will be limiting in the detection of a source's position.

3.4.2 *Overall performance on x and y coordinates*

Given that x and y information of a source is encoded differently in the excitation pattern [31], performance differences in the detection of the x and y coordinate can be expected. In all cases, in noiseless and noisy conditions (see figure 3.6), detection of the source distance y is more reliably determined than its lateral position x.

The MED error is dependent on the x and y coordinate of the sphere, as can be seen in figure 3.8. The error is larger at the left and right edges of the $2D \times D$ field and relatively high when the source is close to the array. These two effects can be explained by the following observations.

The shape of the wavelets as shown in figure 3.2, and thus wavelet ψ, is only partially detected when the source moves close to either side of the field, because the center of ψ is equal to the x coordinate of the sphere. The partially observed wavelets lead to less input information and thus leads to a decrease in performance. Additionally, given the way of sampling training data using trajectories through the sampling space, locations at the borders are slightly underrepresented in train, test and validation data sets.

As can be seen in figure 3.8, the MED increases significantly when the sphere moves close to the sensor array. This result might be partially explained by the distance between the sensors. In Curcic-Blake and Netten [31, pp. 11] it is shown that a source should not be closer than approximately twice the inter-sensor distance to be detected, since the sampling of spatial details in the excitation profile becomes inadequate. For a lateral line with length $2{\cdot}D$, this minimum distance is $\frac{2{\cdot}D}{16} \cdot 2 = 0.25 \cdot D$ for an array of 16 sensors and $\frac{2{\cdot}D}{32} \cdot 2 = 0.125 \cdot D$ for an array of 32 sensors. Figure 3.8 do indeed show decreased accuracy when the y coordinate becomes smaller than these values.

3.4.3 *Noise robustness*

The MLPs show better noise robustness than the ELMs and ESNs. This is at least partially due to the fact that there are less weights present in MLPs. Figure 3.10d shows that a smaller hidden layer size results in better noise robustness for ELMs and ESNs comparable to MLPs, at the cost of lost precision for high signal to noise data. Furthermore, networks with 32 sensors are slightly more robust to noise than networks with 16 sensors.

3.5 CONCLUSION

The current work is one of the very first attempts to investigate the optimal parameter settings for multiple neural network architectures for hydrodynamic imaging using a stationary artificial lateral line. A smaller version of an MLP was previously tested for a sensor array consisting of 6 sensors [4]. Under similar noise levels, but a smaller area, our networks yields a comparable result with an average MED of 2% of D.

3.5.1 *Practical implementation neural networks*

The practical implementation of the neural networks investigated here, when processing flow data measured with an array of velocity sensitive flow detectors, may follow various scenarios. Depending on the particular choice of flow detection of the individual sensors and specifically their signal-to-noise performance using the present results, the optimally performing network may be selected.

To utilize the predictive nature of previously perceived excitation patterns, we explored the use of ESNs containing recurrent connections which outperformed otherwise identical ELMs in noisy conditions. This indicates that memory might help in predicting source locations in simple architectures. Nevertheless, the ESN is outperformed by the MLP, which has no form of (short term) memory. Other types of neural network architectures with recurrent connections may also be helpful in this respect. The network input could for example be extended with excitation patterns of previous time steps or the optional recurrent connections from the output layer to the input layer could be added.

All considered neural network architectures, once trained on simulated data, may respond very fast to velocity input signals of the array and perform localization in real time on standard hardware. In case of the feed forward architectures this is at a rate of several thousand Hz and in case of the ESN several hundred Hz.

3.5.2 *Further research on single source localization*

A specific task that was not trained in the networks investigated so far, is to also estimate the angle ϕ at each time step. The information on ϕ is present in the excitation pattern and provides instantaneous input for estimating the next position.

Extending the input layer with nodes of which the activation is determined by something else than the measured water velocities might also improve the network performance. These indicator nodes can hold information about e.g. the width of the wavelet or the location of minima and maxima in the excitation pattern.

Therefore, other neural networks than considered here also may be considered as alternatives for source localization.

In this research, the network output only gives the 2D location of a source, because it only uses water velocities in the direction parallel to the simulated lateral line. If the excitation pattern along an additional orthogonal lateral line is added to the input, the output can be extended to the location of a moving source in a 3D volume. This orthogonality of neuromast arrays can also be found on the heads and sometimes along the sides of fish [27, pp. 568-576]. On the heads, canals with neuromasts and arrays of superficial neuromasts with different angles to each other are present. Also, fish approach behavior [26] suggests that fish tend to zig-zag towards a source to sense orthogonal projections of flow.

This work cannot readily be used for detecting multiple sources, although it can be altered in at least two ways that might make multiple source localization possible. Firstly, a network for every plausible amount of moving sources could be trained. A problem of this approach is that an accurate criterion for which network output should be used is needed. Secondly, instead of producing the coordinates of the source, the networks could be trained to output a 2D grid in which only nodes that represent locations where a moving source is present have large activation values.

A common additional limitation for source localization algorithms, including these neural networks, is that they assume a stationary artificial lateral line for monitoring sources, rather than a lateral line mounted on a moving body. This latter case requires a different approach since the viscous boundary layer plays an important role [120] and may affect the locally perceived fluid flow [35].

We expect that the use of neural networks will be greatly extended, especially in parallel with the development of alternative and more extended biomimetic velocity sensitive arrays than those reported so far.

4

SIMULATED 1D-SENSITIVE MULTI-SOURCE LOCALIZATION

In this chapter, we explore an alternative approach to object localization that doesn't use regression to directly estimate an objects location. Instead, we use a convolutional neural network (CNN) to reconstruct a heat map from a measured velocity pattern. The maxima in this heat map indicate the location of nearby objects, which allows multiple objects to be encoded simultaneously.

Another expansion of this study is to use two parallel arrays to perform 3D localization. The first step is to reconstruct the heat maps using the measured velocity patterns via the CNN. Then, using an iterative 3D-aware algorithm, we aim to localize up to two objects by combining the information contained in the two reconstructed heat maps.

To assert the performance of the combined system, we report both the reconstruction error of the CNN, as well as the localization error distance (MED) for situations where one object is present, and situations where two objects are present respectively.

PROVENANCE A version of this chapter has been published as:
B.J. Wolf, J. van de Wolfshaar, S.M. van Netten, (2020). Three-dimensional multi-source localization of underwater objects using convolutional neural networks for artificial lateral lines. *Journal of The Royal Society Interface* 17(162) 20190616.

AUTHOR CONTRIBUTIONS BW conceived and implemented the iterative localization algorithm, performed the analysis, wrote the draft, and wrote the final manuscript. JvdW implemented the neural network and helped drafting the manuscript. SvN conceived the study, supervised the study, and revised the draft.

ABSTRACT This research focuses on the signal processing required for a sensory system that can simultaneously localize multiple moving underwater objects in a 3D volume by simulating the hydrodynamic flow as caused by these objects. We propose a method for localization in a simulated setting based on an established hydrodynamic theory founded in fish lateral line organ research. Fish neurally concatenate the information of multiple sensors to localize sources. Similarly, we use the sampled fluid velocity via two parallel lateral lines to perform source localization in 3D in two steps. Using a convolutional neural network, we first estimate a 2D image of the probability of a present source. Then we determine the position of each source, via an automated iterative 3D-aware algorithm. We study various neural network architectural designs and different ways of presenting the input to the neural network; multi-level amplified inputs and merged convolutional streams are shown to improve the imaging performance. Results show that the combined system can exhibit adequate 3D localization of multiple sources.

4.1 INTRODUCTION

Fish are able to accurately sense nearby moving or vibrating objects in water through a lateral line organ [36]. This organ consists of superficial neuromasts (SNs) and canal neuromasts (CNs). The SNs are located externally on the skin and detect the outside flow velocity, whereas the CNs are located underneath the skin in small canals, which filter out low frequencies [75]. As a result, the CNs are sensitive to outside flow acceleration in directions along the canals of the organ. Fish are able to use the mechanosensory input elicited by these neuromasts for a variety of tasks, including tracking down prey [61], or coordinating their movements e.g. in schooling [98].

This biological ability has inspired the development of flow-sensing arrays called artificial lateral lines (ALLs) to perform what is known as hydrodynamic imaging: mapping objects and obstacles via fluid flow interactions. Several ALL set-ups are used for wake detection [22] and source localization [12, 32, 115], usually bench marked by their ability to localize a vibrating dipole source. Most ALL sensor configurations are focused towards a 2D localization problem, while some use out-of-line sensor placement and template-matching methods to localize a phase-locked vibrating source in 3D [127].

The present research contributes to the development of algorithms for ALL hydrodynamic imaging systems as it is the first of its kind to consider localizing multiple objects in three dimensions using machine learning. The field of machine learning might provide the means to tackle these problems, since the resulting systems have exhibited impressive sensory-processing capabilities in a wide range of tasks that require a high degree of generalization, such as computer vision [74] and speech recognition [2].

To feasibly explore a number of different configurations of the artificial neural network, we consider a simulated environment in which objects in motion generate a velocity potential field, which is measured as fluid flow by the ALL. The underlying inviscid hydrodynamic flow model for the local velocity potential is supported by exper-

imental findings in fish lateral line research [31, 43, 121]. In practice, this model may be compromised to create an adequate image of the near surroundings since it neglects vorticity and other possible issues that arise in reality. This demands a system that can cope with a highly dynamic environment and, from the point of view of supervised learning, should generalize well to unseen situations.

In this research, we apply machine learning to explore the applicability of ALL hydrodynamic imaging systems to create local images of moving objects by using simulated ALL inputs. We define the problem as the prediction of a probability function that can be related to the likelihood of a sphere to be in a certain position. This 2D probability grid is then used to produce an estimate of the 3D location using an iterative localization algorithm.

We test several configurations of a convolutional neural network (CNN) to determine the influence of several design choices. While 1D inputs with CNNs have been used before [2, 45], these applications perform dimensionality reduction. In our case, we use the CNN to transform a 1D signal to a 2D probability grid, which requires alterations to a standard CNN architecture. This dimensionality upscaling and specifically its application to the artificial lateral line is completely novel.

In section 4.2, we discuss related research that is relevant to our approach. Section 4.3 elaborates on the details of the 3D source location-encoding model and the general CNN architecture. In section 4.3.4 we discuss the comparative experiments that were conducted to measure the accuracy of our models for locating multiple sources and report the results in section 4.4. Finally, we discuss our findings and list directions for future research in section 4.5 and conclude in section 4.6.

4.2 BACKGROUND

First, we discuss other research on source location encoding with a lateral line organ. Then, we elaborate on the research and insights considering CNNs that inspired the development of our method.

4.2.1 *Source location encoding by the lateral line organ*

Source localization involves arrays of multiple sensors that measure a projection of the local fluid velocity potential in response to an object moving relative to the array.

Findings by Münz [90] and Bleckmann [18] show that the hair cells in each neuromast sensor are aligned in one direction. This causes these sensors to effectively measure a one-dimensional projection of the local pressure gradient as caused by, for instance, a moving object. When these objects are located in the vicinity of the array and they are sufficiently large, the effects of viscosity can be neglected [92]. These findings were used by Curcic-Blake and van Netten [31] to develop a theoretical model and method to compute the pressure gradients between two lateral line canal pores. This model

describes the neural excitation of neuromasts along the lateral line, also known as an excitation pattern. Figure 4.1 illustrates the resulting excitation pattern along a dense neuromast array as generated by a moving sphere.

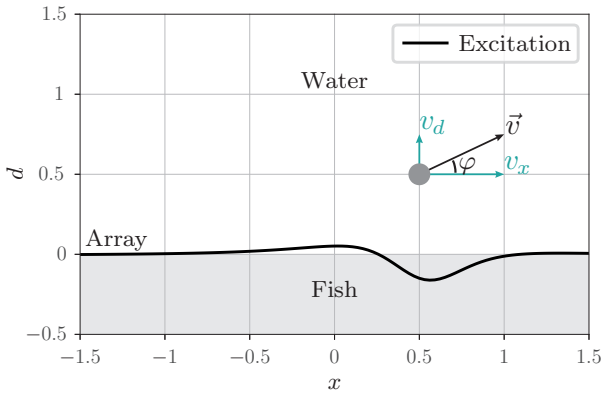


Figure 4.1: Illustration of a situation with a single sphere in a plane with the excitation pattern of a dense neuromast array at $d = 0$.

In two studies [31, 43], it was demonstrated that the information about the location of a vibrating source is encoded in the spatial characteristics of the excitation pattern along the lateral line sensors. This spatial property can then be used to reconstruct a 2D pressure gradient map via the 1D excitation pattern of a single array using wavelet transforms. More specifically, Curcic-Blake and van Netten [31] demonstrate that the excitation of the sensors along the array can be determined from a combination of two wavelets:

$$\psi_{even}(s,x,d) = \frac{1-2\left(\frac{s-x}{d}\right)^2}{\left[1+\left(\frac{s-x}{d}\right)^2\right]^{\frac{5}{2}}}, \tag{4.1}$$

$$\psi_{odd}(s,x,d) = \frac{-3\left(\frac{s-x}{d}\right)}{\left[1+\left(\frac{s-x}{d}\right)^2\right]^{\frac{5}{2}}}, \tag{4.2}$$

where s is the position of the neuromast along the x-axis and (x,d) is the position of the sphere. Here, x denotes the source position relative to the array and d its distance. The actual fluid velocity measured by a neuromast can now be obtained through:

$$v(s,x,d) = \frac{Wr^3}{2d^3}(\psi_{odd}\ \sin\varphi\ -\ \psi_{even}\ \cos\varphi), \tag{4.3}$$

where W is the sphere's absolute velocity, r is its radius, and φ its direction with respect to the array. Note that the left factor of the right-hand side of this equation causes the fluid velocity to scale in a nonlinear way with respect to the distance (d) to the array.

4.2.2 *Object localization using artificial lateral lines*

Traditionally, template-matching methods have been used to create a 2D heat map [115] and in some cases a 3D volume [127] of a single source. More recently, artificial neural networks have been used for single source localization using both physical [4, 5] and simulated [20, 122] sensor arrays.

4.2.2.1 *2D localization*

For 2D localization of a single sphere, a classical lateral line geometry is often used, where all sensors are positioned equidistantly on a single line. Using the inviscid hydrodynamic model described in the previous section, one can readily determine the velocity potential at each of the sensors in this ALL from the relative location and motion of a source $S^{(i)}$: the forward problem.

$$F(S^{(i)}) = u_{\text{obs}} \tag{4.4}$$

The concatenation of these locally measured fluid velocities u_{obs} at each sensor is known as a velocity pattern, analogous to the excitation pattern. The effective problem statement that ALL setups face is the inverse problem; from the measured velocity pattern, reconstruct the relative position and motion of a source.

$$F^{-1}(u_{\text{obs}}) = S^{(i)} \tag{4.5}$$

This inverse problem (F^{-1}) has been tackled via template matching and beamforming methods [115]. These methods use a library of modeled or measured velocity patterns and a correlation scheme to retrieve the source location of a newly presented velocity pattern.

An example of Capon's beamforming is described in detail by Pandya et al. [97]. Here, the authors make use of an outer-product correlation matrix between the sensors for a single time step. The second key component is a vast 3D library (x, d, φ) of modeled velocity patterns for a range of possible source locations and orientations. This matrix and library are then used to create a heat map with Capon's method. The predicted location is finally determined by finding the maximum in this heat map.

In addition to these beamforming methods, non-convolutional neural networks have been used for 2D localization. A relatively small multi-layer perceptron (MLP) with one hidden layer having 24 nodes was applied to an artificial lateral line array consisting of 6 sensors in a row [4, 5]. There, the position of a vibrating sphere was varied and reconstructed in a 2D plane. Boulogne et al. [20] assessed the localization performance for a single source in a 2D plane for three different types of neural networks: the MLP, an echo state network (ESN), and an extreme learning machine (ELM), where the latter proved to be optimal. This type of network was also used in [121] to localize both moving and stationary vibrating sources in a 2D plane. Recently [123], the ELM architecture was

compared to a recurrent network architecture (LSTM) for objects moving in a straight line in a 2D plane. To the authors' knowledge, the present work is the first effective demonstration of a CNN architecture for localizing a source with an ALL.

4.2.2.2 *3D localization*

The problem space around the classic artificial lateral line geometry is circle symmetric; one cannot discriminate distance in the y plane vs distance in the z plane. Other sensor geometries that break this symmetry are required to extend this problem to 3D localization.

Yang et al. [127] introduced a cross geometry on a cylinder, where 9 1D-sensitive sensors were positioned in a straight line with 3 sensors perpendicular at either side of the center of the array. To localize the source in a 3D volume around the lateral line setup, they extended the beamforming algorithm [97] to work with a 5D library containing the (x, y, z) location and orientation (azimuth and zenith angles). As in [97], the location of the source was selected from the maximum in the 3D heat map volume. A slightly different approach is needed for localizing multiple sources.

Van de Meulen et al. [88] localized two sources, positioned in a simulated 3D basin with several geometries of 16 sensors placed at the bottom of this basin. There, an artificial neural network was tasked to reconstruct a 3D heat map volume using the sampled fluid velocity at each sensors location. Yet, because the chosen geometries were spatially uncorrelated, the network couldn't make use of the spatial properties encoded in velocity patterns as sampled in a line.

Our chosen sensor geometry of two parallel lines with equidistant sensors retains the ability to make use of the spatial excitation pattern properties, while allowing 3D localization. These spatial properties make this problem well suited to be learned by convolutional neural networks.

4.2.3 *Convolutional neural networks*

Convolutional neural networks (CNNs) have received increasing attention in machine learning research in recent years. These networks are especially suited at 2D data driven tasks for images such as the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in image classification, detection, and visual segmentation tasks [56, 74, 104, 107]. Other than computer vision, CNNs have also been applied to 1D signals in speech recognition [2] and haptic tactile classification [45].

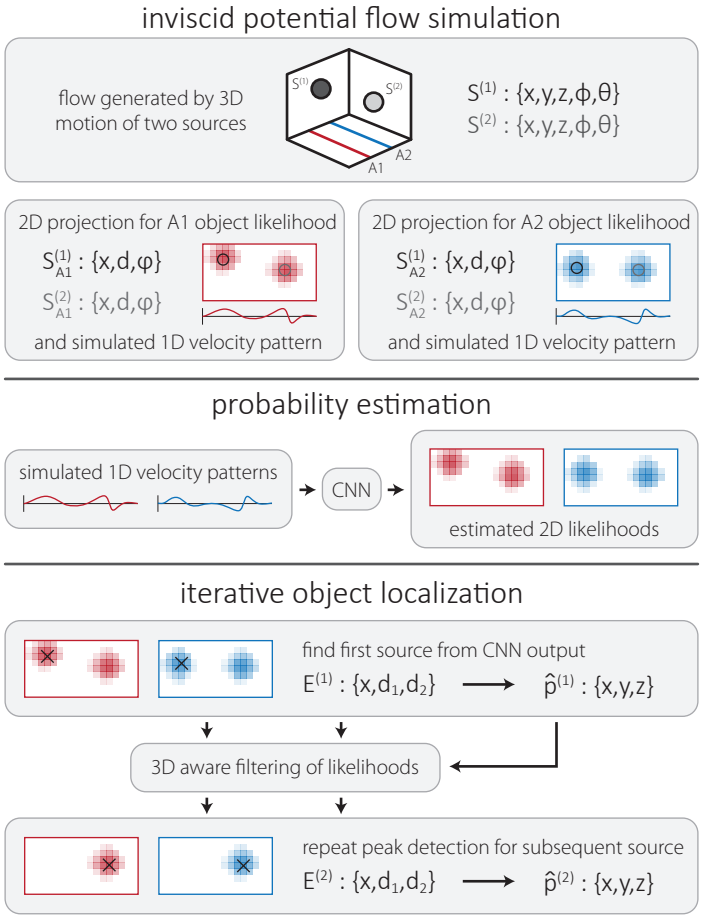


Figure 4.2: Schematic representation of the task and processing pipeline. The simulation generates examples of objects in motion, accompanied by a 2D projection of object probability and the 1D velocity pattern at each array. The overall task is to localize these objects using the 1D velocity patterns as input in two steps: estimating the 2D probability grids of a source present and then determining the coordinates of the object in 3D using these grids.

CNNs differ from the more standard multi-layer perceptrons [103] in their more efficient architectural design and were initially proposed in their current form by LeCun et al. [79]. A key insight behind the development of this kind of network architecture is that the pixels in an image are spatially correlated. This means that meaningful features of an image such as edges, corners and color transitions can be found in small sub regions. This allows specialized feature representations (filters) to be tuned more easily, since they are trained via the larger collection of sub regions rather than the whole image.

A CNN is usually made up from several convolution layers, interlaced with pooling layers to reduce the dimensions of the convolutional stream. Typically, additional convolutional layers converge to filters that hierarchically extract more abstract features [130]. In the extreme case, an image can be reduced via several convolutional and pooling layers to a single output neuron for binary classification.

Our alterations to the standard implementation of the CNN are further described in section 4.3.3.

4.3 METHODS

This section elaborates on source location encoding in 3D, the exact simulation implementation, the global architecture of the CNN used and its tested variations, and the algorithm for iterative location decoding in 3D. An overview of the whole process is shown in figure 4.2.

4.3.1 *Location encoding in 3D*

We first consider the situation of a single sphere in a 2D plane after which we extend it to multiple spheres and a 3D volume.

4.3.1.1 *Spheres in a plane*

With a single sphere in a plane, the object motion is described by a speed W and direction φ. When a sensor array is positioned at $d = 0$, we can determine the fluid velocity at each sensor s using the source's position (x, d) via the fluid flow model as described by equation (4.3).

With the location of the source encoded in the velocity pattern, we can localize the source by effectively decoding this profile. If only a single sphere is considered, localization can be treated as a regression problem in which an inverse method predicts the components of the position vector $\boldsymbol{p}$. However, if the number of spheres is arbitrary, the problem cannot be formulated as a regression problem anymore. Instead, we replace

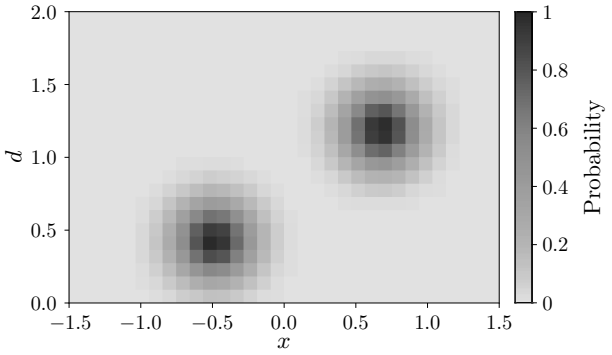


Figure 4.3: Plot of the target probability function in a 2D plane for 2 sources. The target function is evaluated at (x, d) where x is taken from 32 equidistant points between −1.5 and 1.5 and d is taken from 22 equidistant points between 0 and 2.0.

the target regression values by a probability function f, as illustrated in figure 4.3, that is defined over a plane:

$$f(\boldsymbol{a}) = \max_i \exp\left(-\frac{\|\boldsymbol{p}^{(i)} - \boldsymbol{a}\|^2}{2 \cdot \rho^2}\right), \tag{4.6}$$

in which $\boldsymbol{p}^{(i)}$ represents the position vector of the ith sphere and ρ is a smoothing factor. The vector $\boldsymbol{a}$ is any coordinate for which we want our model to predict the likelihood of a source present.

We thus have two instances of the 2D target probability function, one for each array, for which we define the following discrete domain for array A:

$$\begin{aligned} D_A = \Big\{ & (x, d_A) \mid \\ & x \in \{-1.5, -1.5 + \delta_x, \ldots, 1.5\}, \\ & d_A \in \{0, \delta_d, \ldots, 2\} \Big\}. \end{aligned} \tag{4.7}$$

Figure 4.3 depicts a discretized example target function for two sources in a single plane. We choose the domain for x to be from −1.5 to 1.5 in 32 parts and for d_A from 0 to 2 in 24 parts. This fully encompasses the bounded space of motion for the source (see section 3.2.1).

4.3.1.2 *Moving spheres in 3 dimensions*

In order to extend the problem to three dimensions, we consider a planar projection for each spherical source that is defined by the location of sensor array A and the $\mathbb{R}^3$ position vector $\boldsymbol{p}$ of the source.

To conveniently work out the planar projection of source motion with respect to the sensor array, we construct the source's velocity vector $\boldsymbol{v}$ in $\mathbb{R}^3$. Next, we project this source velocity $\boldsymbol{v}$ on the plane spanned by the array A and the source position $\boldsymbol{p}$ to determine a projected velocity vector $\boldsymbol{v}_A$.

We know that the velocity component parallel to the array A, assuming the array is aligned with the x-axis, is simply $\boldsymbol{v}_{Ax} = \boldsymbol{v}_x$.

The orthogonal velocity component $\boldsymbol{v}_{Ad}$, requires taking into account both the y and z components of the position and velocity vector. This orthogonal component is spanned along $\boldsymbol{p}_{yz} = [0 \quad p_y - A_y \quad p_z]^\top$ and is thus given by:

$$\boldsymbol{v}_{Ad} = \boldsymbol{v} \cdot \frac{\boldsymbol{p}_{yz}}{\|\boldsymbol{p}_{yz}\|}. \tag{4.8}$$

The third velocity component that is orthogonal to the projected plane has no contribution to the measured fluid velocity at the sensor array and is therefore neglected for this study.

From the parallel and orthogonal components of the projected velocity vector $\boldsymbol{v}_A$, it is trivial to obtain the absolute velocity W_A and angle φ_A. Finally, we find $d_A = \sqrt{(\boldsymbol{p}_y - A_y)^2 + \boldsymbol{p}_z^2}$ and substitute the projected parameters in equations (4.1), (4.2), and (4.3); fully defining the 1D velocity pattern for array A for any source in a $\mathbb{R}^3$ volume.

4.3.1.3 *Resolving 3D ambiguity*

If we only use a single sensor array, we can estimate the x_A coordinate and distance d_A of a source with respect to array A. This evidently causes ambiguous situations where the source could be anywhere in a ring around this array, since the y and z coordinate of the object are combined in d_A.

By adding a second array, we have two instances of the probability function, one for each sensor array. Both instances map to toroidal shapes that intersect at the source's target position as indicated in figure 4.4, thereby collapsing the ambiguity. For simplicity and without loss of generality, we will assume that the first array, A_1 is placed at $y = -0.5$ while the second array, A_2 is placed at $y = 0.5$.

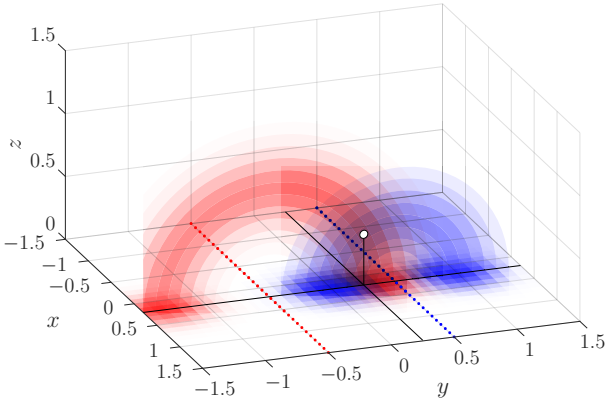


Figure 4.4: The planar probability functions of both arrays are ambiguous in a torus around their array. Shown here is a cross section of the probability tori for a single source. The intersection of maximal likelihood corresponds to a source's location in $\mathbb{R}^3$.

4.3.2 *Data synthesis*

We simulate spherical sources moving within a bounded $\mathbb{R}^3$ space in a basin. This bounded space is fully encompassed by the discretized domain D_A.

4.3.2.1 *Source motion*

For our data synthesis, we consider a simulated environment in which multiple spheres move through a basin measuring $3 \times 3 \times 1.5$ m^3. A source is allowed to move in a bounded space with $x, y \in [-1, 1]$ and $z \in [0, 1]$. Each sphere moves with a constant speed W of 0.05 m·s^{-1} and their direction of motion is described by two angles (ϕ, θ) where ϕ is the azimuthal angle in the x, y-plane and θ is the polar angle with respect to the positive z-axis. The direction of each sphere changes according to:

$$\begin{aligned} \phi_{t+1} &= \phi_t + \text{unif}[-1\,\text{rad}, 1\,\text{rad}], \\ \theta_{t+1} &= \theta_t + \text{unif}[-1\,\text{rad}, 1\,\text{rad}]. \end{aligned} \tag{4.9}$$

In case the source is about to move outside the bounded space, we let it ricochet as in [20]. Here, the reflected angle is the same as the incident angle.

At any point in time, at least one and at most two spheres can be in the basin. Spheres disappear with a probability of 5% each time step while reappearing at a random location with the same probability. This results in a data set where one source is present 53% of the time, while in other cases, two sources are present.

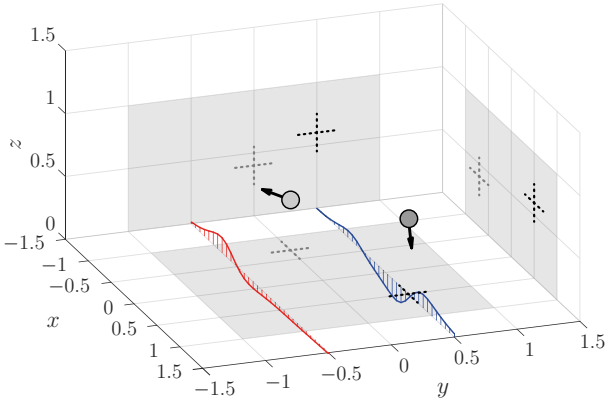


Figure 4.5: Illustration of a situation with two spheres in a 3D environment together with the velocity patterns of two arrays in the bottom plane aligned with the x-axis. The grey backdrops indicate the motion bounds and source location with cross hairs.

4.3.2.2 *Velocity pattern sampling and noise*

The velocity patterns are sampled from two simulated sensor arrays placed at the bottom of the basin. We obtain these velocity patterns via the fluid model described in section 4.2.1. For each sensor array, we can simply sum the contributions from each sphere, because of the properties of the assumed potential flow.

We then introduce relative noise on top of these sampled velocity patterns, via a noise level parameter n. This allows us to make fair comparisons of the network between different signal-to-noise levels. The parameter n expresses a ratio of the input's (i.e. the water velocity) standard deviation, denoted $\sigma_\forall$. The computation of this standard deviation considers the velocity patterns for all time steps and all sensors in both arrays. The noise is added by sampling from a normal distribution with mean 0 and a chosen variance of $n \cdot \sigma_\forall$.

4.3.3 *CNN implementation*

The neural network receives 1D velocity patterns of both sensor arrays and is tasked to approximate two probability grids, one for each array. Table 4.1 and figure 4.6 provide an overview of the default CNN architecture, which is further discussed in the following subsections.

Table 4.1: Overview of the default neural network architecture. Where two kernel sizes are listed, each is used for half of the kernels. When the streams are merged, the kernels concatenate so that the total number of kernels per layer is unaffected.

LAYER DESCRIPTION	OUTPUT MERGED/SPLIT	KERNELS PER STREAM	KERNEL SIZE	ACTIVATION FUNCTION
Input	Split	τ	NA	x
Conv. 1	Split	32	5, 7	$\tanh(x)$
Conv. 2	Split	64	5, 7	$\mathrm{ReLU}(x)$
Conv. 3	Merged	2×64	5, 7	$\mathrm{ReLU}(x)$
Conv. 4	Merged	2×64	5, 7	$\mathrm{ReLU}(x)$
Output layer	NA	24	5	$\mathrm{sigmoid}(x)$

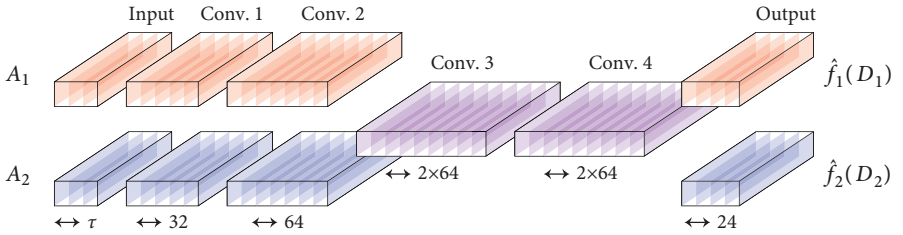


Figure 4.6: Illustration of the default network architecture. The color of each layer indicates to which stream it belongs. In this case, the 3rd and 4th layer are merged to allow the two streams to exchange information.

4.3.3.1 *Input mapping*

The magnitude of the sampled velocity patterns can have a considerable dynamic range, given the cubed relation with the distance to the source, see equation (4.3). To normalize and map this dynamic range of the input, we squeeze the activation at the first hidden layer in the range of $[-1, 1]$ by using the hyperbolic tangent function.

As a first input augmentation, we consider feeding both attenuated and amplified versions of the input to the first hidden layer before normalization. The hyperbolic tangent function shows differences in its output more clearly when its input is around 0. Therefore, using different amplification levels simultaneously will allow the network to capture the dynamics at different magnitudes more adequately than a system with a single amplification level.

As an alternative augmentation to the inputs, we also consider an aspect of time-delay neural network architecture that effectively introduces history. In the network

architecture, we vary a time delay frame parameter, τ. When $\tau = 1$, we only use the current velocity pattern for each array, so no effective history is added. In case of $\tau > 1$, the input is constructed via concatenating velocity patterns of previous time steps. It is hypothesized that using past velocity patterns will help the model to interpret otherwise ambiguous situations during testing. Moreover, it might also help to reduce the impact of input noise.

From this point forward, we consider the amount of sensors that sample a velocity pattern as the width of the input, i.e. 32. The additional past velocity patterns that are added as effective history are considered depth slices of the input.

4.3.3.2 *Convolutional streams*

The velocity patterns from both arrays are sampled at equidistant locations. This geometry introduces a spatial relation between the shape of the velocity patterns and the position of the source (see section 4.2.1). The convolutional layers in this neural network architecture are especially suited to parse this type of data, since these layers effectively perform 1D convolutions that can make use of this spatial relation.

The convolutional layers are used in combination with the Rectified Linear Unit (ReLU) activation function: $\mathrm{ReLU}(x) = \max(0, x)$ [48]. The usage of the ReLU activation function reduces the vanishing gradient problem and provides sparsity to the feature representations in the network, which in turn helps for linear separability of features within a layer [48].

To maintain the local spatial relations throughout the convolutional stream, only convolutional layers are used, as opposed to regular CNNs which also incorporate pooling layers. The series of convolutional layers combine to *convolutional stream*s, one for each sensor array. These streams start at the input (index 0) and end at the output layer (index 5). Both streams have the identical task to map an input velocity pattern to its probability grid as the desired output. Hence, we can use the exact same weights for each convolutional stream that predicts one of the two probability functions. In other words, we can share the convolutional streams' weights. Sharing the weights results in reusing the same kernels in both streams. Disabling sharing allows separate filter kernels to be trained for each convolutional stream.

The convolutional layers all have a width of 32 (just as the input) and a stride of 1. This width is maintained by zero padding the input of each convolution at both ends. The depth of the convolutional layer determines the number of filter kernels.

In addition to weight sharing, we can also *merge* both convolutional streams. This creates a single stream that is responsible for predicting the probability output for both sensor arrays. The streams are merged by concatenating the outputs of hidden layers at a certain layer index, which doubles the layer depth from that point on. In our experiments, we vary this layer index, shifting the merging point. After this merging point, the subsequent layers will receive information from both sensor arrays.

4.3.3.3 *Output mapping*

For each sensor array and accompanying convolutional stream, we have one target probability function. At the end of the convolutional stream, two different methods are used and compared to map the activation to the output layer. By only using convolutions, we have maintained local spatial relations; this enables us to use the last convolutional layer directly for this mapping. We also investigate replacing the last (output) layer with a fully connected layer. This type of layer may uncover patterns that stretch over a wider part of the sensor arrays, at the increased risk of over fitting. In both cases, the output layer has a width of 32 and a depth of 24, reflecting the chosen discretization of the domain D_A.

Since our target output has values in the range $[0, 1]$, we decided to use a sigmoid function $\varsigma(x) = 1/(1 + \exp(-x))$ at the output layer, which ensures the output to be in that range.

4.3.4 *CNN optimization*

This section describes the default CNN settings, how we train the CNN, and the chosen architecture variations for these settings.

4.3.4.1 *Optimization criterion*

The convolutional neural network is trained to estimate two 2D discretized probability functions based on two sampled velocity patterns, one from each array. The network minimizes a custom loss function that takes into account the true and predicted probability as well as a weight regularization term. The exact definition of the loss function that we minimize is:

$$\begin{aligned} \mathcal{C}(A, \boldsymbol{a}) = \hat{f}_A(\boldsymbol{a}) \cdot \log f_A(\boldsymbol{a}) + \\ (1 - \hat{f}_A(\boldsymbol{a})) \cdot \log(1 - f_A(\boldsymbol{a})) \end{aligned} \tag{4.10}$$

$$\mathcal{S}(A) = \sum_{\boldsymbol{a} \in D_A} -\mathcal{C}(A, \boldsymbol{a}), \tag{4.11}$$

$$\mathcal{W}(\boldsymbol{w}) = \frac{1}{2} \sum_i \boldsymbol{w}_i^2, \tag{4.12}$$

$$\mathcal{L}_{\text{total}} = \mathcal{S}(A_1) + \mathcal{S}(A_2) + \lambda \mathcal{W}(\boldsymbol{w}). \tag{4.13}$$

Equation (4.10) shows the binary crossentropy loss for a single position $\boldsymbol{a}$ for array A, where $f_A(\boldsymbol{a})$ is the actual probability and $\hat{f}_A(\boldsymbol{a})$ is the predicted probability for said position. The binary cross-entropy loss has the desirable property that, when combined with sigmoid activations, its gradients are of approximately the same magnitude across all possible preactivation values.

Equation (4.11) shows the loss for a single array A, which simply sums the losses over all positions in the corresponding domain D_A.

Equation (4.12) is a regularization term in which $\boldsymbol{w}$ is a vector that contains all trainable parameters of the neural network. This term penalizes large weight vectors, making the whole network less prone to over fitting [94].

Finally, equation (4.13) shows the total loss for a single sample (i.e. a single input and output pair) in which λ governs the contribution of the regularization term for the trainable weights. In this study, we use batches of 64 samples for updating the CNN weights.

Table 4.2: Default model and hyper parameters

PARAMETER	SYMBOL	VALUE
First sensor array	A_1	$y = -0.5$
Second sensory array	A_2	$y = 0.5$
Sphere radius	r	0.05
Smoothing factor	ρ	0.2
Relative noise level	n	0.001
Input range factors		[1000]
History length	τ	3
Merging index		3
Weight sharing		enabled
Output layer		FCNN
Learning rate	η	1×10^{-3}
Regularization coefficient	λ	1×10^{-4}

4.3.4.2 *CNN variations*

Our experiments are designed to characterize the influence of design decisions that would be relevant for an artificial lateral line system with convolutional neural networks. The parameter settings for the default model are listed in table 4.2. We consider the following variations to the default model:

- Noise level: we assess the influence of noise on the final performance of the system and choose $n = 0.0001, 0.001, 0.01, 0.02$.
- Amplification levels: we compare a system with a single input amplification level of 1000 with another system that uses the concatenated input of three input

amplification levels: 100, 1000 and 10,000. This causes the latter system to have three times as many input depth slices as the former.

- History length: we compare different history sizes $\tau = 1, 2, 3, 4$ to see whether the system benefits from integrating the inputs of multiple time steps.
- Merging index: we vary the merging index from 0, where the input of both arrays is concatenated and parsed by a single stream, to index 5 where no merging takes place.
- Weight sharing: for the unmerged layers, we can choose to share the weights by reusing the same kernels in both convolutional streams, or disable sharing so that we use separate kernels for each stream.
- Output layer: we compare a network that has a convolutional output layer, forming a fully convolutional neural network (FCNN) with a network that has a fully connected output layer (CNN+FC).

4.3.4.3 *Validation*

We create five different pairs of train and validation data, each generated independently using the model as discussed in section 4.3.2. For each of these pairs, the train data contains 10 000 samples, while the validation data contains 2000 samples.

4.3.5 *Location decoding in 3D*

Finally, we estimate $\mathbb{R}^3$ positions $\hat{\boldsymbol{p}}$ from the CNN output. Using the output 2D probability grids, we present a 3D-aware algorithm to detect both the number of sources and their location (see figure 4.7). This algorithm benefits from being 3D-aware, since there could be two distinct objects that have a similar distance to one of the arrays, effectively masking each other in 2D. This algorithm is described here in detail using an example; a formal description in the form of pseudocode may be found in the supplementary material at the end of this chapter.

We first detect the x-coordinate of the highest probability, by flattening and adding both probability grids, and taking the maximum of the resulting probability line (white '+' in figure 4.7A,B). Then, for each array, we find the d coordinate with the maximal value for that x-coordinate and fit a Gaussian (see equation 4.6) near the maximal coordinates to yield $(\hat{x}, \hat{d})$ coordinate pair estimates (black '×' in figure 4.7A,B). From the estimates $\hat{d}_1$, $\hat{d}_2$ and the array y-coordinates, it is trivial to work out the $\hat{y}$ and $\hat{z}$ estimate for the first source, completing the first position vector.

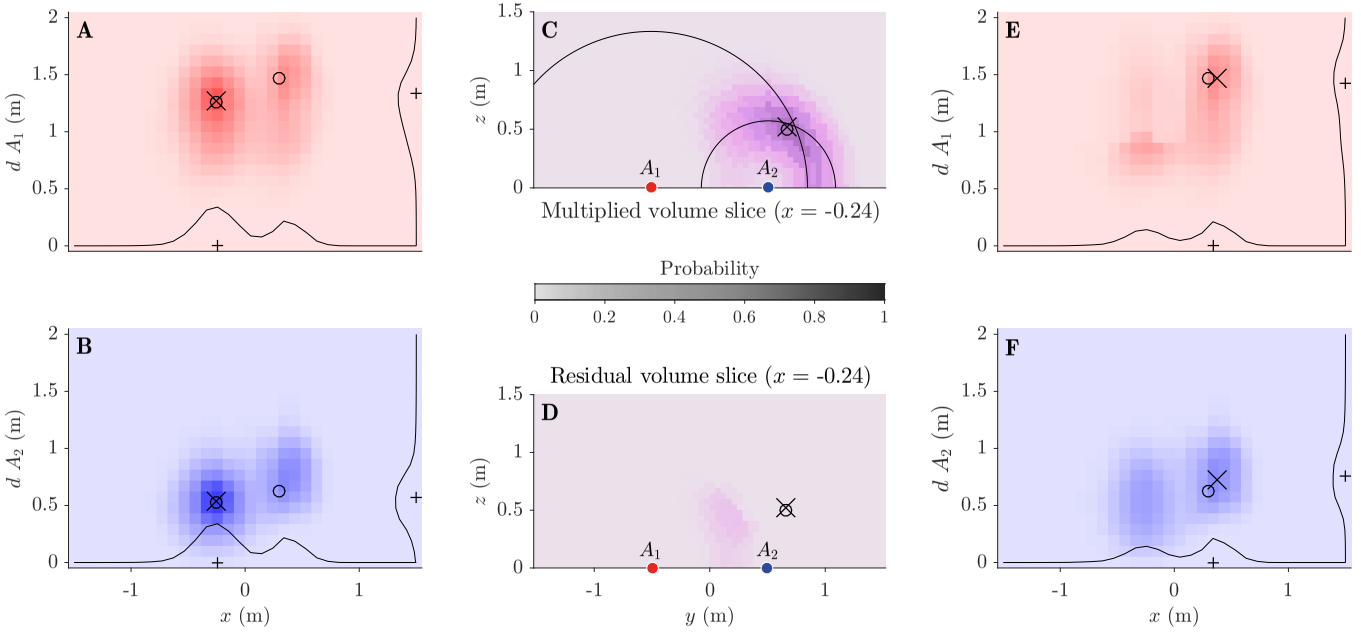


Figure 4.7: Steps for location decoding in 3D. The circles denote the source locations, the crosses show the reconstructed locations, and the pluses indicate discrete maxima for the probability. Panes A and B show the CNN output probability estimation, which is used to locate a first source. Pane C shows one slice of the volume resulting from projecting the probability grids to 3D and multiplying the volumes resulting from A_1 and A_2 respectively. The arcs correspond with the 'plus' symbols for $d\ A_1$ and $d\ A_2$. Pane D shows a slice of the residual volume after 3D filtering. Finally, panes E and F show the residual probability grids, constructed from the residual volume.

The next step is to create probability volumes for each reconstructed grid, by effectively rotating these grids, as indicated in figure 4.4. We choose a voxel size of $(\delta_x, \delta_y, \delta_z) = (\frac{3}{31}, \frac{3}{63}, \frac{3}{63})$ and fill each voxel with the value of the nearest mapping to 2D coordinates. We use element-wise multiplication to combine both array's probability volumes to a single volume. A slice of this volume is shown in figure 4.7C.

To remove the estimated probability values from the already found source, we calculate two probability grids from $\hat{x}$, $\hat{d}_1$, and $\hat{d}_2$ using the target function from equation (4.6) and similarly create a probability volume. We then subtract this probability volume from the prior and map the residual volume (figure 4.7D) back to two residual probability grids (figure 4.7E,F). This remapping consists of two steps. First, each pixel in the grid is filled with the maximal value found from the inverse rotational mapping. Then, for each value in the remapped probability grids, we take the square root, since these values originates from a multiplication.

We then repeat the first step of finding $(\hat{x}, \hat{d})$ coordinate pair estimates. As is made visible in figure 4.7E and 4.7F, the 3D filtering method also leaves residual probability at the position of the first source. We, therefore, use two thresholds to determine whether a second detection constitutes a second source. First, the summed probability of the residual probability grids from the last step should be higher than 55, the expected value for target probability grids containing a single source. And secondly, its estimated position should be further than 0.55 m to the first object's estimated position.

As an error metric for object localization, we report the localization error in MED (m) per object. For samples where we expect two sources, we report how many sources were detected per sample.

4.4 RESULTS

As described in section 4.3.4.2, several types of variations for the CNN architecture were considered, each given a different color in upcoming figures. We both asses the loss and accuracy with regards to reconstructing the target probability functions, as well as the localization error from the iterative detection algorithm using these reconstructions.

4.4.1 *Probability grid reconstruction*

In terms of CNN training loss (table 4.3), the default hyperparameter settings seem to produce the best performing network in terms of 2D probability reconstruction. The fact that the loss values are relatively large is a result of summing the binary cross-entropy per location in equation (4.11), rather than taking the average.

An example of the CNN output for two sources can be seen in figure 4.7A and 4.7B. Here, and in most other samples, the reconstructed probability grids don't show perfect

Table 4.3: Target probability reconstruction metrics. The default setting entry (D) is repeated where appropriate.

	LOSS	MSE	$\pm\,\sigma$
Default	142.3	5.28×10^{-3}	5.99×10^{-4}
$\tau = 1$	143.0	4.83×10^{-3}	3.71×10^{-4}
$\tau = 2$	143.5	5.25×10^{-3}	4.90×10^{-4}
(D) $\tau = 3$	142.3	5.28×10^{-3}	5.99×10^{-4}
$\tau = 4$	143.8	5.33×10^{-3}	6.57×10^{-4}
$0.0001 \times \sigma_\forall$	142.4	4.84×10^{-3}	3.80×10^{-4}
(D) $0.001 \times \sigma_\forall$	142.3	5.28×10^{-3}	5.99×10^{-4}
$0.01 \times \sigma_\forall$	147.6	6.07×10^{-3}	6.83×10^{-4}
$0.02 \times \sigma_\forall$	148.5	6.35×10^{-3}	6.91×10^{-4}
Merge at 0	153.3	7.03×10^{-3}	6.20×10^{-4}
Merge at 1	151.3	6.86×10^{-3}	8.53×10^{-4}
Merge at 2	148.4	6.46×10^{-3}	8.29×10^{-4}
(D) Merge at 3	142.3	5.28×10^{-3}	5.99×10^{-4}
Merge at 4	146.0	5.58×10^{-3}	5.36×10^{-4}
No merging	156.3	7.57×10^{-3}	8.42×10^{-4}
Multi-range	143.7	4.35×10^{-3}	1.99×10^{-4}
No sharing	143.4	5.41×10^{-3}	7.09×10^{-4}
Fully connected	166.9	8.40×10^{-3}	3.29×10^{-4}

Gaussians, but the maxima in these grids often coincide with the actual position of that object.

There are however four variations that outperform the default settings in terms of the average probability reconstruction MSE. The first two variations ($\tau = 1$ and $\tau = 2$) have a shorter history length. Given the standard deviations, the effect of this parameter on the reconstruction quality is marginal and provides no significant improvement. The third variation to outperform the default settings is the lowest relative input noise setting ($0.0001\sigma_D$). This is expected, as the emulated sensors effectively have a higher sensitivity. The final outperforming variation (multi-range) concatenates amplified and attenuated versions of the input for the CNN input. While the training loss is slightly higher than default, this variation performs significantly better in terms of reconstruction quality.

4.4.2 *3D position reconstruction*

Using the reconstructed 2D probability grids from the CNN, we used the iterative algorithm explained in section 4.3.5 to detect sources and determine their position in 3D. Figure 4.7 shows the process for a single sample for determining the position of two objects. It shows the advantage of taking our approach for detecting maxima in these fields.

The first estimates for simply picking the maximum pixel (indicated with pluses) is a good start, but only provides estimates from a discretized coordinate system. As is more clearly visible in figure 4.7C (the intersection of the two arcs), this initial estimate tends to overestimate the distance. While the final position from the fitting procedure does not necessarily coincide with the ground truth source positions, these final estimates do clearly reflect the maxima of the probability grids.

Figures 4.8 and 4.9 show the localization error distribution for a hypothetical perfect probability reconstruction and the reconstruction from each type of CNN model alteration. The performance of these model alterations, broadly speaking, align with the quality of probability grid reconstruction.

For detecting the location of a single source (figure 4.8), the default model has the lowest median error, but the shorter tail of the 'multi-range' model suggests that its performance is more consistent. All models perform reasonably well, given that the average random prediction error in a bounded volume of $2 \times 2 \times 1$ m is 1.132 m. Especially the default model and the 'multi-range' model result in a localization performance that is close to optimal.

In the cases where two sources are present (figure 4.9), the performance of the localization algorithm is only slightly affected in the ideal case. For the CNN probability grid reconstructions, the differences between most CNN models decrease, while the localization error per source more than doubles overall. With respect to the model alterations, most trends visible in the case with one object are also visible here, with one notable exception. Here, the 'multi-range' model outperforms the default model, signified by its lower median and its shorter third quartile.

In some cases, the pipeline described in section 4.3.5 was not able to reliably determine a second source. The default CNN model has an object detection rate of 85.4%. Only the $\tau = 1$, 'Merge at 1', and 'multi-range' variants have a detection rate that is on par.

4.5 DISCUSSION

Here, we discuss the overall results, followed by the effects of the extensive range of architectural design variations for the CNN, as well as the effectiveness of the incorporated localization method. This is followed by a discussion on the detection limits of the approach and directions for future research.

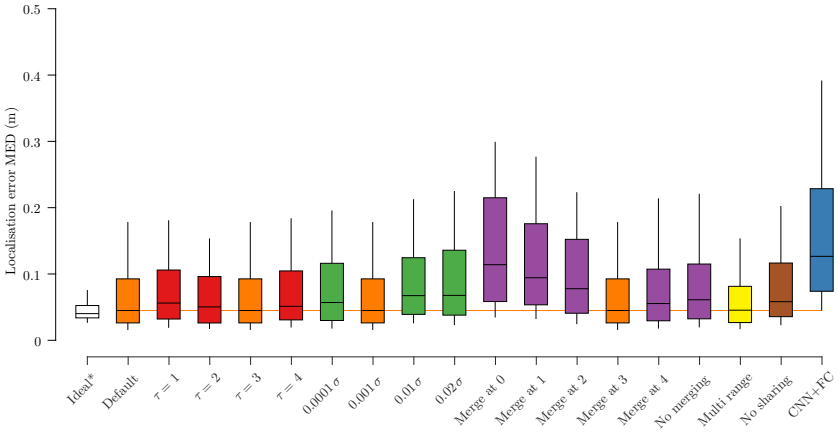


Figure 4.8: Localization error for samples containing a single source. Indicated here are the 10%, 25%, 50%, 75%, and 90% error distribution. The performance of the default model is repeated where appropriate. All sources were found in this subset of the data set. The Ideal* case reflects the performance on a hypothetical perfect probability grid reconstruction.

4.5.1 *The processing pipeline*

The final average error in localization performance can be thought of as a cascading effect of imperfect maxima localization, which is influenced by an imperfect probability reconstruction, which can be caused by unbalanced sampling in the simulation or a suboptimally configured CNN. Uncertainty or information loss can be introduced and interact at each of these stages.

While better probability reconstructions tend to result in better localization performance overall, optimizing these intermediate steps does not guarantee optimizing the end result. This is made apparent by the reconstruction performance of a CNN with $\tau = 1$, which is the best variant in terms of MSE. It is however outperformed in localization by a CNN with $\tau = 3$. In this case, the values in the reconstructed probability grids of the CNN with $\tau = 1$ are closer to the ground truth on average, but their maxima may have been further away from the actual position than to those in the case with $\tau = 3$.

4.5.2 *CNN design variations*

There are a number of parameters that were not optimized in detail such as the learning rate, the regularization loss, the number of hidden layers and the number of neurons per hidden layer. It is likely that our system can be improved by performing a grid search, random search or evolutionary search to optimist these hyperparameters [15, 17]. We discuss here the effects of the parameters that were varied.

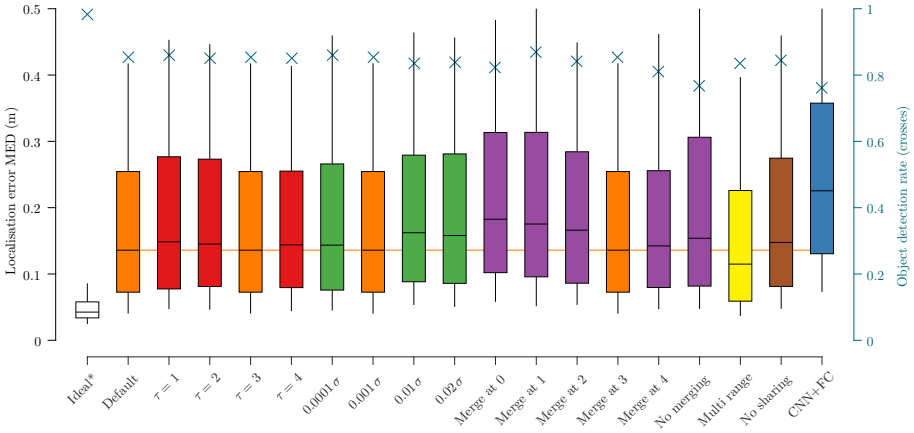


Figure 4.9: Localization error for samples containing a double source. Indicated here are the 10%, 25%, 50%, 75%, and 90% error distribution for the left axis. The right axis and crosses denote the object detection rate, where 1 indicates both sources are found and 0 indicates none are found. The Ideal* case reflects the performance on a hypothetical perfect probability grid reconstruction.

HISTORY τ. First of all, the history length indicated by the τ parameter seems to have a marginal effect on the reconstruction performance. Perhaps surprisingly, a history length of $\tau = 1$ results in the most accurate model. This could be explained by the fact that it is easier to generalize from single time steps than it is to generalizing from multiple time steps.

Perhaps in a more complex setting it might still be valuable to incorporate previous time frames, as has been shown for localizing a single source using regression [123]. One such situation that may benefit from taking history into account is when objects turn more slowly than the current maximal 1 rad (57 degrees) per second or when the objects can vary their speed. These more complex situations would likely reduce the performance of the system as currently optimized. It would require more training data to be generated for tuning any inverse method for localizing these objects.

NOISE. There seems to be no obvious relation between the noise level and localization performance. We see that when the noise level is 10 times lower than the default, we obtain a marginally worse performing model. The performance with higher noise levels follows intuition and also degrades. It is likely that some form of noise may have helped preventing overfitting for the default CNN architecture.

MERGING CONVOLUTIONAL STREAMS. Similarly, the default CNN model may have the optimal merging index. We note that the number of neurons is kept the same for any merging index, and thus should not cause a difference the performance.

Merging at the input (index 0) leads to the worst average localization performance. In this case, the streams are not separated at all. As we postpone the merging point, we initially observe a gradual improvement. However, after merging index 3, we observe impeded performance. Perhaps this could be explained by the fact that the earlier layers will be able to extract the local characteristics of the velocity patterns of a single sensor better when not affected by the velocity patterns of the other array. This makes it easier for the first few layers to generalist the representation that they learn, as the number of possible inputs per layer is greatly reduced by splitting the two streams.

In the case of merging index 4, there is no hidden layer between the merged streams and the output. This prevents the CNN for making use of nonlinear processing, which is likely to be beneficial. This might also explain why we see a relatively poor performance when no merging is used at all.

AMPLIFICATION LEVELS. Using multiple amplification levels results in a considerable improvement compared to the default model for probability function estimation and localization, especially when two sources are considered.

The cubic term in equation (4.3) causes the input to be in a large dynamic range. By using multiple amplification levels, the network can encode salient differences in sensor inputs at multiple scales, which improves the overall performance. This effect is more prominent in samples containing two objects. In these cases, both near (strong) and far (weak) sources are likely to be processed in a favorable dynamic range in one of these amplification levels.

WEIGHT SHARING. Disabling weight sharing between the streams seems to have a negligible effect on performance. Perhaps if more arrays were aligned, weight sharing may be more beneficial, since it makes training more effective.

ADDING A FULLY CONNECTED LAYER. It seems that using a fully connected layer at the back of the network, instead of a final convolution layer, severely impedes performance. Since the amount of trainable weights is increased considerably, it is likely that this is a result from overfitting. Perhaps with a bigger data set, the CNN+FC may still prove to be useful.

4.5.3 *Iterative source detection*

The 3D-aware filtering method provides an iterative approach to detecting multiple objects as detected by multiple arrays. A consequence of the 3D aware filtering method is that, after removing the most prominent source, there is some residual probability

left on that location (see figure 4.7D). In our case, via reasonably chosen thresholds, the expected summed probability for a single source and a minimal distance, we were able to detect a second object in most cases. There are two main advantages of this method.

Firstly, objects which are masked in 2D, can still be detected. It could happen that two objects are positioned at (0, 1, 1) and (0, 0, 1) respectively. These two sources would have and identical distance to A_2, and thus sensed as a single object by that array, but they can be discerned and detected as separate objects via array A_1.

Secondly, without specifying how many sources should be detected, this algorithm can detect an arbitrary number of sources. As it is the first demonstration of localizing an arbitrary number of objects with an ALL, we cannot readily compare this method with other (regression) inverse methods in the literature without adjusting them.

We note that the current data generation does allow for two sources to be spawned at the exact same place, with no minimal distance enforced. This has in some cases led to a single perceived object, which affected the detection rate. For future simulations and experiments, a minimal distance between objects may improve the quality of probability reconstruction and the source detection performance.

4.5.4 *Detection limits and future research*

Hydrodynamic imaging is a near-field modality. The distance range in which this near-field localization method can be used is limited. The hydrodynamic information carried through the water deteriorates with a the cube of the distance d^3, while for (far-field) sound signals this is a factor of the distance squared d^2, which is the case for sonar. While the detection range for sonic detection is bigger, hydrodynamic detection can be more effective in the near-field.

Scaling up this principle is therefore not trivial, as hydrodynamic imaging is mostly beneficial in the near-field range. This near-field can be extended by increasing the source's speed and volume. We could, for instance, scale up the source properties, setup, and domain with a constant factor. However, even when the hydrodynamic signals can be amplified to counter the effect of an increased distance, the resolution of sampling a velocity pattern will drop and likely impair the system. At further distances, sonar-based solutions have a clear advantage.

The application of hydrodynamic detection of objects can therefore not scale up indefinitely, but the range can be extended beyond the current chosen domain. It has been shown that the source can be positioned in an area next to the array and still be detected reliably using different types of artificial neural networks [6, 121, 123]. It is, therefore, not necessary for the whole velocity pattern to be sampled within the area directly in front of the array; the domain can extend beyond the length of the array. The CNN might be less effective in situations where only a part of the velocity pattern is detected, since some of the spatial characteristics are not sampled.

The current iterative method of localization performs equally well in situations of one and two objects present, and is therefore not a bottleneck in this pipeline. However, even in the ideal case where the method is presented with a perfect probability reconstruction, some errors remain. The lower bound on this error might be a result of the unbiased sampling of locations; in some cases, two objects were instantiated in nearly the same location.

For future research, it should therefore also be investigated whether the current iterative position reconstruction method is the most appropriate in cases with non-overlapping or even more objects. Other peak-detecting algorithms may be equally able to cope with the 2D masking in probability grids constructed from 1D velocity patterns resulting from objects moving in 3D space.

Our research can be extended by further increasing the variation on the input. One could increase the number of simultaneous objects and allow different sizes, and different shapes, and a variable speed. It is evident that the more variations the system has to learn to cope with, the more data will be required to train the system. The results obtained in this research suggest that there is potential for our implementation to address such complicated scenarios.

4.6 CONCLUSION

In this chapter, we assessed a new approach to underwater object localization using a simulated ALL with convolutional neural networks. Via the iterative 3D-aware position estimation algorithm, we have shown that multiple spheres can be detected simultaneously in a 3D space by using two sensor arrays placed in parallel.

The most significant CNN model improvement is a result of providing the input at several amplificiation levels. Since the input was compressed and normalized between -1 and 1, these additional amplification levels effectively made the CNN more compatible with the considerable dynamic range of the input. Additionally, by placing two ALLs in a parallel configuration, the two convolutional streams of the CNN could be merged and exchange information, which is beneficial for localization performance.

We have therefore demonstrated that the combined system is suitable for localizing multiple moving objects in a bounded 3D volume.

SUPPLEMENTARY INFORMATION

The iterative 3D aware localization algorithm, as explained in section 4.3.5, is described here with pseudocode. This description indicates the overall structure of the iterations in algorithm S4.1. The three non-trivial sublevel algorithms are also described in algorithms S4.2, S4.3, and S4.4 respectively.

Algorithm S4.1: iterative localization algorithm

DATA: probability grids $f_1(D_1)$ and $f_2(D_2)$.
RESULT: position vector $\hat{\boldsymbol{p}}$ per detected source.

$H_1 \longleftarrow f_1(D_1)$
$H_2 \longleftarrow f_2(D_2)$

DO

return a location
$\hat{x}, \hat{y}, \hat{z} \longleftarrow$ FINDSOURCE(H_1, H_2)
$\hat{\boldsymbol{p}} \longleftarrow \hat{x}, \hat{y}, \hat{z}$

filter found source in 3D
$V_{\text{cur}} \longleftarrow$ MAP2VOLUME(H_1, H_2)
$H_{\text{est1}}, H_{\text{est2}} \longleftarrow$ 2DGAUSS$(\hat{x}, \hat{y}, \hat{z})$
$V_{\text{est}} \longleftarrow$ MAP2VOLUME$(H_{\text{est1}}, H_{\text{est2}})$
$V_{\text{res}} \longleftarrow V_{\text{cur}} - V_{\text{est}}$
$H_1, H_2 \longleftarrow$ VOLUME2MAPS(V_{res})

determine stopping criterion
totEnergy $\longleftarrow \Sigma H_1 + \Sigma H_2$

WHILE *totEnergy* ≥ 55

Algorithm S4.2: findSource

INPUT : 2D maps H_1, H_2
OUTPUT: position estimate $\hat{x}, \hat{y}, \hat{z}$

find coordinates of maximum value
$E_x \longleftarrow$ MAXCOORD$(\Sigma_x(H_1) + \Sigma_x(H_2))$
$E_{d1} \longleftarrow$ MAXCOORD$(H_{1,x=E_x})$
$E_{d2} \longleftarrow$ MAXCOORD$(H_{2,x=E_x})$

fit equation (6) near initial estimate position
$\hat{x_1}, \hat{d_1} \longleftarrow$ FIT2DGAUSS(H_1, E_x, E_{d1})
$\hat{x_2}, \hat{d_2} \longleftarrow$ FIT2DGAUSS(H_2, E_x, E_{d2})

transform x, d estimate to R3 position
$\hat{x} \longleftarrow$ MEAN$(\hat{x_1}, \hat{x_2})$
$\hat{y}, \hat{z} \longleftarrow$ TRANSFORM$(\hat{d_1}, \hat{d_2})$

Algorithm S4.3: map2volume

INPUT : 2D maps H_1, H_2
OUTPUT: 3D volume V

FOR *every coordinate* $\boldsymbol{c} \in$ *volume* V DO

determine distance to each array
$d_1 \longleftarrow \sqrt{\boldsymbol{c}_z{}^2 + (\boldsymbol{c}_y + 0.5)^2}$
$d_2 \longleftarrow \sqrt{\boldsymbol{c}_z{}^2 + (\boldsymbol{c}_y - 0.5)^2}$

find map values on nearest 2D coordinate
$V_1(\boldsymbol{c}) \longleftarrow H_1(\langle \boldsymbol{c}_x \rangle, \langle d_1 \rangle)$
$V_2(\boldsymbol{c}) \longleftarrow H_2(\langle \boldsymbol{c}_x \rangle, \langle d_2 \rangle)$

return element-wise multiplied volume V
$V \longleftarrow V_1 \odot V_2$

END

Algorithm S4.4: volume2maps

INPUT : 3D volume V
OUTPUT: 2D maps H_1, H_2

$H_1, H_2 \longleftarrow$ ZEROES

FOR *every coordinate* $\boldsymbol{c} \in$ *volume* V DO

determine distance to each array
$d_1 \longleftarrow \sqrt{\boldsymbol{c}_z{}^2 + (\boldsymbol{c}_y + 0.5)^2}$
$d_2 \longleftarrow \sqrt{\boldsymbol{c}_z{}^2 + (\boldsymbol{c}_y - 0.5)^2}$

find nearest 2D coordinates to write
$H_1(\langle \boldsymbol{c}_x \rangle, \langle d_1 \rangle) \longleftarrow \text{MAXVAL}(H_1(\langle \boldsymbol{c}_x \rangle, \langle d_1 \rangle), V(\boldsymbol{c}))$
$H_2(\langle \boldsymbol{c}_x \rangle, \langle d_2 \rangle) \longleftarrow \text{MAXVAL}(H_2(\langle \boldsymbol{c}_x \rangle, \langle d_2 \rangle), V(\boldsymbol{c}))$

return element-wise square rooted maps H_1, H_2
$H_1 \longleftarrow \sqrt{H_1}$
$H_2 \longleftarrow \sqrt{H_2}$

END

5

SIMULATED 2D-SENSITIVE LOCALIZATION

This chapter directly compares an array of 1D-sensitive flow sensors versus an array of 2D-sensitive sensors in simulation, to determine whether 2D-sensing provides an advantage. In addition, we compare two methods of data generation: a grid-wise location method and a simulated motion path.

We adapt the simulation space slightly compared to those used in the previous chapters; the bounded area of interest where the object may be positioned now extends beyond the length of the array. We then use an ELM to estimate the coordinates and the direction of motion, or orientation, of the object within this area of interest.

We report the localization error distance (MED) and the orientation estimation error (degree) for both methods of data generation. For both these error measures, we study the spatial distribution of errors, indicating the effect of object distance on the estimation errors. Finally, we study the influence of sensor noise on the system, to show which data method and which type of sensor array may be more robust with respect to noise.

PROVENANCE This chapter has been previously published as:
B.J. Wolf, S.M. van Netten, (2019). Training submerged source detection for a 2D fluid flow sensor array with extreme learning machines. *Proc. SPIE 11041, Eleventh International Conference on Machine Vision (ICMV 2018)* 1104126.

AUTHOR CONTRIBUTIONS BW conceived the study, implemented the simulation, performed the analysis, wrote the draft, and wrote the final manuscript. SvN revised the draft.

ABSTRACT An array of fluid flow sensors can be used to detect and track underwater objects via the fluid flow field these objects create. The sensed flows combine to a spatio-temporal velocity profile, which can be used to solve the inverse problem; determining the relative position and orientation of a moving source via a trained model. In this study, two training strategies are used: simulated data resulting from continuous motion in a path and from vibratory motion at discrete locations on a grid. Furthermore, we investigate two sensing modalities found in literature: 1D and 2D sensitive flow sensors; all while varying the sensor detection threshold via a noise level. Results show that arrays with 2D sensors outperform those with 1D sensors, especially near and next to the sensor array. On average, the path method outperforms the grid method with respect to estimating the location and orientation of a source.

5.1 INTRODUCTION

The lateral line is a mechanoreceptive organ found in most fish on the head and trunk, which enables them to detect nearby objects and obstacles [36]. This organ consists of distributed discrete sensors called neuromasts, which detect local fluid flow relative to the fish body [18]. This additional sense, sometimes referred to as 'touch at a distance' augments the fish perception and allows behaviors such as schooling and prey detection [28].

Artificial lateral lines (ALLs) are arrays of fluid flow sensors and can be used to help autonomous underwater vehicles (AUVs) safely navigate in murky waters [117], as this sense does not rely on vision. Alternatively, the array could be mounted on a stationary platform, such as on piers and in harbors, to monitor sub-surface traffic or for instance large (schools of) fish. Next to visual inspection and sonar, this passive sensor system provides an undetectable method that works in the near field and does not require light or an active beacon.

The common challenge ALLs have to solve is the inverse problem: from the measured fluid velocities at discrete locations, reconstruct the source location, size, speed, and direction. Assuming a hydrodynamic model is available, a data set can be built (the forward problem), which encodes a source's state in a velocity profile. However, this model does not directly provide a solution to decode the velocity profile; thus introducing the inverse problem.

5.1.1 *State of the art*

Several methods for solving the inverse problem have been put forward for both simulated and physical ALLs, with a focus on localization. They are usually bench marked by their performance on localizing a sphere vibrating either parallel (x) or orthogonal (y) to the array. In Yang et al. [125], a training set was created by placing a sphere vibrating in the x-direction on roughly 400 locations in a $2L \times 0.5L$ grid, where L is the length of

the array. Using a Gaussian mixture model, they were able to reconstruct the location of a subset with an average error of 0.5% L. In [4], a $L = 10$ cm array with six equally spaced 1D sensors was used to localize a sphere vibrating in both the x- and y-direction. Here, using a $2L \times L$ grid of <100 samples, a multi-layer perceptron (MLP) with 24 hidden nodes was able to localize a source with a maximum Euclidean error of 3% L.

Boulogne et al. [20] used simulated motion to add realistic variation in the orientation of the source. An MLP, echo state network (ESN), and extreme learning machine (ELM) were used to localize a source moving in a path. Here, the ELM architecture outperformed the MLP and ESN architectures in both accuracy (0.2% L) and execution time. The MLP architecture, although more powerful by design, has more tunable hyper-parameters. This means that an MLP takes significantly longer to optimize in practice. Similarly, the ESN, making use of a dynamic reservoir which functions as a short term memory, did not benefit from an effective history under low noise conditions. We therefore choose the ELM architecture to compare two data generation methods: a vibrating object at discrete locations versus object motion in paths.

These two methods have not been compared before in terms of their resulting performance. In practice, a vibrating object needs to be repositioned and produces a very clean signal which might not be representative for what the ALL will sense in response to nearby moving sources. The alternative method could be practically implemented via tracking a fish, or controlling a robotic fish, and using the tracked object location and orientation as training data for a machine learning or template matching algorithm.

Furthermore, while most artificial lateral line sensors are sensitive to a single (usually parallel, x) component of the fluid flow, some use alternating sensor directions such that they are sensitive to the orthogonal or y-component of fluid flow. In a recent study, 2D sensitive ALL sensors were introduced [124].

5.1.2 *Aims of this study*

The goal of this study is to find the optimal method for training an algorithm for an ALL using a limited size data set, without over fitting on the data. For this we select an ELM artificial neural network. Simulated data is used to control both the data-set size, and vary the noise level; increasing the applicability of the results to a wider range of hydrodynamic events.

In the present study, the two different strategies are compared: grid, where a stationary vibrating source is positioned on grid points in several directions, and path, where a source moves through the simulation space. These strategies will be compared via their influence on the performance of determining the location and orientation of a moving source. In addition, two types of sensors are compared: those only sensitive to the parallel velocity component, 1D sensors; and those sensitive to both velocity components, 2D sensors.

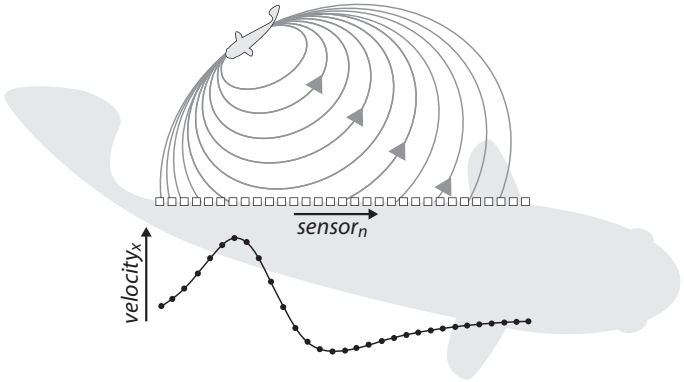


Figure 5.1: Fish lateral line perception. The smaller fish creates a flow field which is sensed by the biological flow sensors on the fish body. The resulting sampled excitation pattern encodes the source's location and other properties.

5.2 METHODS

A schematic representation of the fish lateral line is depicted in figure 5.1. Here, the small moving fish source creates a dipole field which is sampled at each sensor's location. From this picture, it is clear that the sampled velocity profile, or excitation pattern, along the fish trunk holds information about the source's lateral position (i.e. x-coordinate). In fact, the velocity profile uniquely encodes the lateral position, the distance, and orientation of a source [31, 43]. These profiles are well described by dipole fluid flow models, assuming that a source is moving with constant velocity [43] or vibrating in a particular direction [5, 31, 50]. This allows generating data sets for vibrating and moving objects in an area of interest. Figure 5.2 shows examples of such simulated velocity profiles.

5.2.1 *Setup and velocity profiles*

The simulation environment has an area of interest of size $2L \times L$, where L is the length of the array. The center of the bottom edge marks the origin, as indicated in figure 5.2. In this area of interest, a spherical object with diameter $0.1L$ is tracked by the artificial lateral line.

The 16 sensors are placed $0.1L$ below the bottom edge of this area between the coordinates $(-0.5L, -0.1L)$ and $(0.5L, -0.1L)$. To determine the velocity profiles as measured

by this array, a near-field simplified potential flow fluid model [92] is used. This model describes the radial v_r and tangent v_θ flow components in polar coordinates:

$$v_r = a^3 \cdot r^{-3} \cdot w \cdot \cos(\theta - \varphi), \tag{5.1}$$

$$v_\theta = {}^1/_2 \cdot a^3 \cdot r^{-3} \cdot w \cdot \sin(\theta - \varphi), \tag{5.2}$$

where w is the instantaneous velocity of the source, φ its direction or orientation, and a its radius. These radial and tangential contributions are then used to form the parallel v_x and orthogonal v_y velocity profiles:

$$v_x = v_r \cdot \cos\theta - v_\theta \cdot \sin\theta, \tag{5.3}$$

$$v_y = v_r \cdot \sin\theta + v_\theta \cdot \cos\theta. \tag{5.4}$$

Several examples of these 2D velocity profiles are listed in figure 5.3. When the source is positioned further from the array, the amplitude of the velocity profile decreases, which decreases the signal to noise ratio. In addition, the velocity profiles undergo spatial broadening, as indicated by the dashed lines. The distance of the source is therefore encoded in both the signal to noise ratio and spatial broadening [20, 31]. Varying the x-coordinate of the source laterally shifts the velocity profiles. The source angle φ with respect to the array is encoded in the ratio of mother wavelets [31] in the sampled velocity profile. While motion parallel $\varphi = \{0, 2\pi\}$ and orthogonal $\varphi = \pm\pi/2$ to the array produce odd and even mother wavelets respectively, other source directions result in a mix. This causes features such as the maxima or zero-crossings of the velocity profiles to shift; they are therefore less indicative of the source's position.

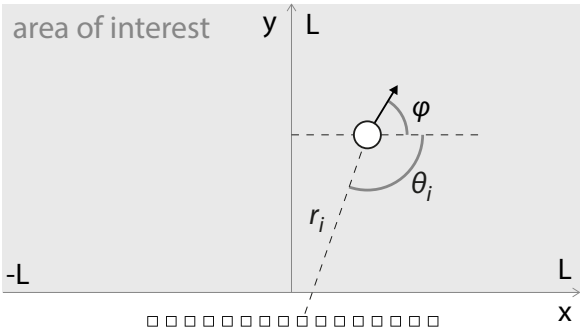


Figure 5.2: Schematic representation of the setup. Here, the sensor array (indicated with squares) measures velocity profiles caused by the moving white sphere. The local fluid velocity for each sensor i is determined via the angle of object motion with respect to the array φ, the distance of said object r_i, and relative angle θ_i.

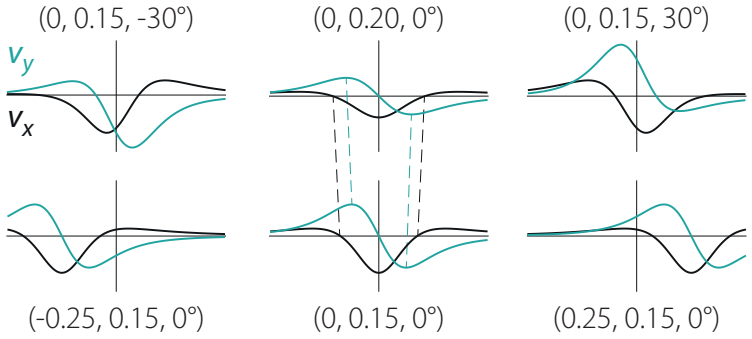


Figure 5.3: Example calculated velocity profiles for several states (x, y, φ). The bottom row displays the lateral translation property for the velocity profiles. The top corners show variations in movement direction, which causes both profiles to change. The center top example shows both effects of increased distance: reduced amplitude and the broadening of spatial features.

5.2.2 *Data set strategies*

Examples of the two different strategies found in literature are shown in figure 5.4. The grid strategy divides a $2L \times L$ area into a grid with a fixed distance between grid points. On each grid point, the source is oriented in k random directions, resulting in a data set of length $k(2m^2 - m)$, where m is the number of points in length L.

The path strategy [20] initializes a source at a random location and orientation within the area of interest. Then, per time step, the source moves a fixed distance l in a random direction in the range $\varphi_t = \varphi_{t-1} + \text{rand}[-\pi/4, \pi/4]$. To keep the source within the area of interest, we alter the motion near the area boundaries. When the source is less than a turning circle radius away from an edge, the source direction is changed with maximally $\pm\pi/4$ per step.

5.2.3 *Preprocessing and normalization*

The velocity profiles are usually sampled by sensors with finite precision. To model this finite precision, noise is added to the velocity profiles, which represents a threshold velocity: the lowest detectable fluid velocity for each sensor. This added noise is sampled from a uniform distribution $[-\eta, \eta]$. As a preprocessing step, each noisy sampled velocity profile $v^*[n]$ is then normalized between [-1, 1] as in Boulogne et al. [20] by

$$v_*[n] = v_*[n] \,/\, \max|v_*[n]|. \tag{5.5}$$

This process removes absolute velocity magnitudes, but information about the source is inherently present in both the signal to noise ratio of the velocity profile and the

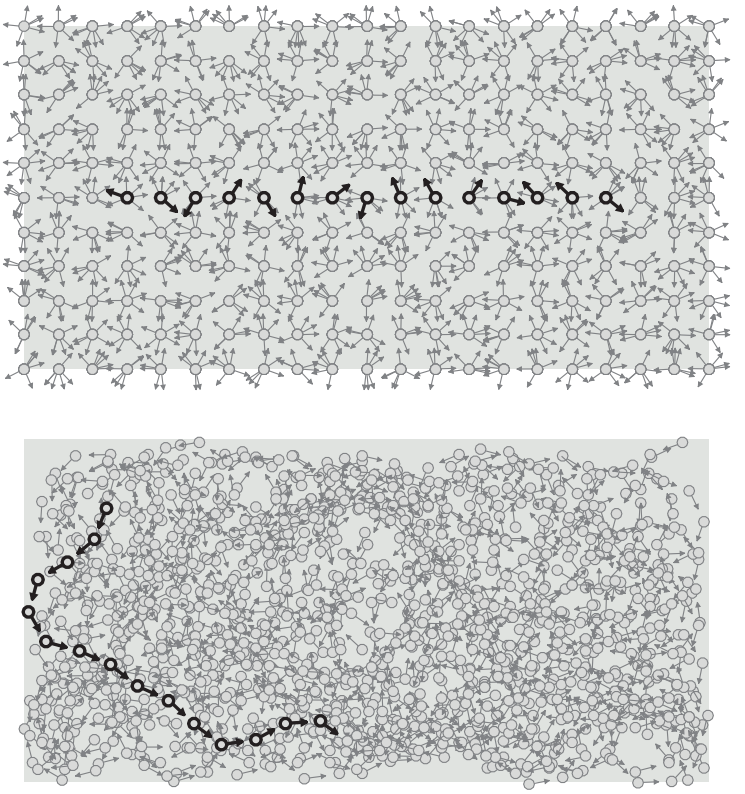


Figure 5.4: Subsequent source locations and orientations for grid (top) and path (bottom) data generation strategies.

spatial features such as the zero crossings, maxima, and velocity profile broadening (see section 5.2.1).

The output of the network is also normalized. For the source locations, both x- and y-coordinates are all within [-1, 1], so this does not cause a normalization problem. For the source orientation, the value for φ can have a large range and this angle representation raises a wrap-around problem around 0 and 2π. By taking the sin φ and cos φ as a two-output representation, this wrap-around problem is conveniently solved and ensures that the output is within [-1, 1].

In the case of 1D sensors, only the sampled v_x velocity profile is used for the data set. In the case of the 2D sensors, the sampled v_x and v_y profiles are concatenated after normalization.

5.2.4 *Extreme learning machine*

The extreme learning machine (ELM) is a single layer feed forward artificial neural network. The input layer fully connects to the single hidden layer with randomly initialized weights, which are not altered further. The ELM can be trained in a fraction of the time of e.g. an MLP, because only the fully connected hidden-to-output weights are learned in one step via a pseudo inverse matrix [64].

In our case, the input layer corresponds to normalized noisy velocities as sampled at each sensor's position. The input-to-hidden weights are fully connected and initialized from a uniform distribution [-0.5, 0.5], they are not altered during training. Depending on the sensor type, the number of input nodes is either 17 (16 sensors + 1 bias) in the case of 1D sensors, or 33 in the case of 2D sensors.

The activation function $f(x) = \tanh(x)$ is used to calculate the hidden layer output. In order to perfectly map the hidden output H from a data set to the desired outputs T, a weight matrix W can be found such that $HW = T$, where the teacher matrix T contains the corresponding x, y, $\cos\varphi$, and $\sin\varphi$ target values. To train this network, a least squares solution for W is found via the MoorePenrose generalized pseudo inverse of H [64]: $W = H^{\dagger}T$.

This process determines $H^{\dagger}$ and therefore the optimal hidden weights W very fast. In addition, this artificial neural network architecture has only a single tunable hyper parameter (see section 3.1), which makes it a suitable architecture for performing a large number of comparative experiments.

5.2.5 *Performance measures*

The ELM minimizes the mean squared error (MSE) for all four network outputs. However, this averaged MSE is not an intuitive measure for source detection performance. Therefore, two different performance measures are reported for a trained network. The location estimation error is quantified via the mean Euclidean distance (MED) between the true x, y source locations and the network predictions for the whole test set. The orientation estimation error is quantified using the true and estimated angle φ from the $\cos\varphi$ and $\sin\varphi$ network outputs. For an intuitive performance measure, we use the mean absolute difference in degrees.

5.3 EXPERIMENTAL PROCEDURE

5.3.1 *Hyper-parameter optimization*

The ELM has only one hyper-parameter to optimize for the experiments, the hidden layer size. To prevent over fitting on the training data, a separate validation set is used. ELM hidden layer sizes were selected on a logarithmic scale between 50 and 2000 in

16 steps. For each network size, 5 neural networks were trained and tested on newly generated data sets with aforementioned settings.

For this task, a 1D sensor grid training data set was used with $m = 11$ grid points per length L and $k = 6$ orientations per grid point. Its performance was measured on a 1D sensor path validation set of length 4000 with step size $l = 0.1$. With a source speed of $w = 0.5L$ m/s and source radius $a = 0.05L$ m, the lowest sampled velocities are in the order of $10^{-6}L$ m/s. Therefore, a noise level of $\eta = 10^{-6}L$ m/s was chosen for this optimization. This noise level is varied during the comparative experiments.

5.3.2 *Influence of strategy, sensor type, and noise*

With the optimal hidden layer size for the grid strategy and 1D sensor type combination, networks with both strategies and sensor types were trained in order to determine its effect on the source detection performance. For the path strategy, an equal length training data set is used. The performance of each network was measured on a novel path test set of length 4000 with step size $l = 0.1$, to allow a fair comparison.

The path strategy is expected to enhance performance on the orientation estimation. The simulated motion through the simulation space might produce more relevant location-orientation pairs in the data set for this task. However, this strategy introduces biases into the training set, which might impair generalization. The grid strategy may likely produce less performance variance, especially near the boundaries of the area of interest.

Furthermore, 2D sensors are expected to be beneficial for both localization and orientation estimation performance, compared to 1D sensors. The two velocity profiles complement each other; where one profile has a zero-crossing, the other will have a non-zero velocity magnitude (see figure 5.3). In addition, the combination of the two readings per sensor reduces the effect of noise.

Noise levels for the experiments are selected from a logarithmic scale, and used to determine its effect on the performance of all four combinations of strategy and sensor type. Since the noise-level affects the signal to noise ratio of the velocity profiles, an increase in noise is likely to decrease the detection range and therefore performance in the area of interest.

5.4 RESULTS

5.4.1 *Hyper-parameter optimization*

A grid training set with 1386 samples was used, such as depicted in Figure 5.4 (left), to find a suitable hidden layer size for the ELM. In general, the number of weights should be less than the data set size to prevent over fitting. Figure 5.5 indicates that both location

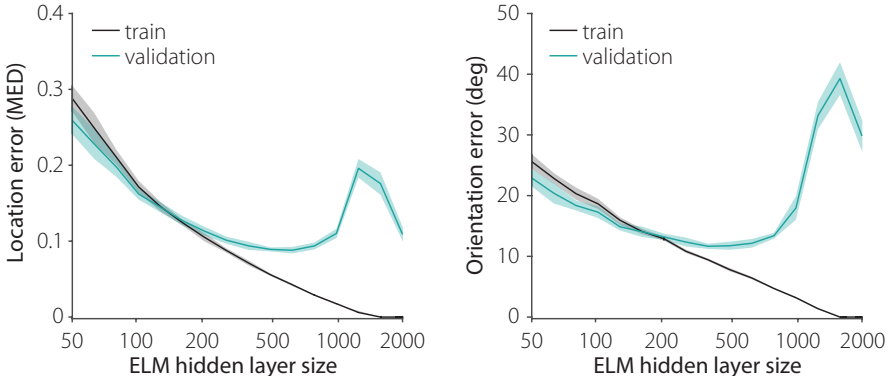


Figure 5.5: Averaged (N=5) performance (lines) and standard deviation (fills) for different hidden layer sizes.

and orientation estimation have the lowest error on the unseen validation set around a hidden layer size of 400; this size is used for the comparative experiments.

5.4.2 *Influence of strategy and sensor type*

Figure 5.6 shows an example (with noise $\eta = 10^{-6}L$ m/s) of the averaged interpolated location and orientation estimation error in a $2L \times L$ area for four combinations of strategy and sensor type. The overall performance, regardless of strategy and sensor type, does show variation in the area of interest. In addition, the error increases in regions close to the array and at further distances from the sensors.

Furthermore, the addition of the orthogonal velocity profile v_y increases the performance and has a greater influence on the estimation error than the data set strategy. Yet, the path strategy does outperform the grid strategy in most regions, as is visible from the darker hue in the location based error plots (Figure 5.6).

The distribution of the estimation error for this example, including the median and variance, are also reflected by the box plots in Figure 5.7. These box plots all indicate a long tailed distribution towards higher errors, with the median relatively low. For both sensor types, the median and third quartile (Q3) for the path strategy indicate that, on average, path outperforms grid. Especially for the orientation estimation performance, both the strategy and sensor type have a significant effect on the error distribution.

5.4.3 *Influence of noise*

Figure 5.8 shows that both the location and orientation estimation error reach their minimum near $\eta = 10^{-7}L$ m/s. Furthermore, the difference in performance between the

four cases increases with decreasing noise-levels. The standard deviations for both types of errors indicate that the performance is consistent.

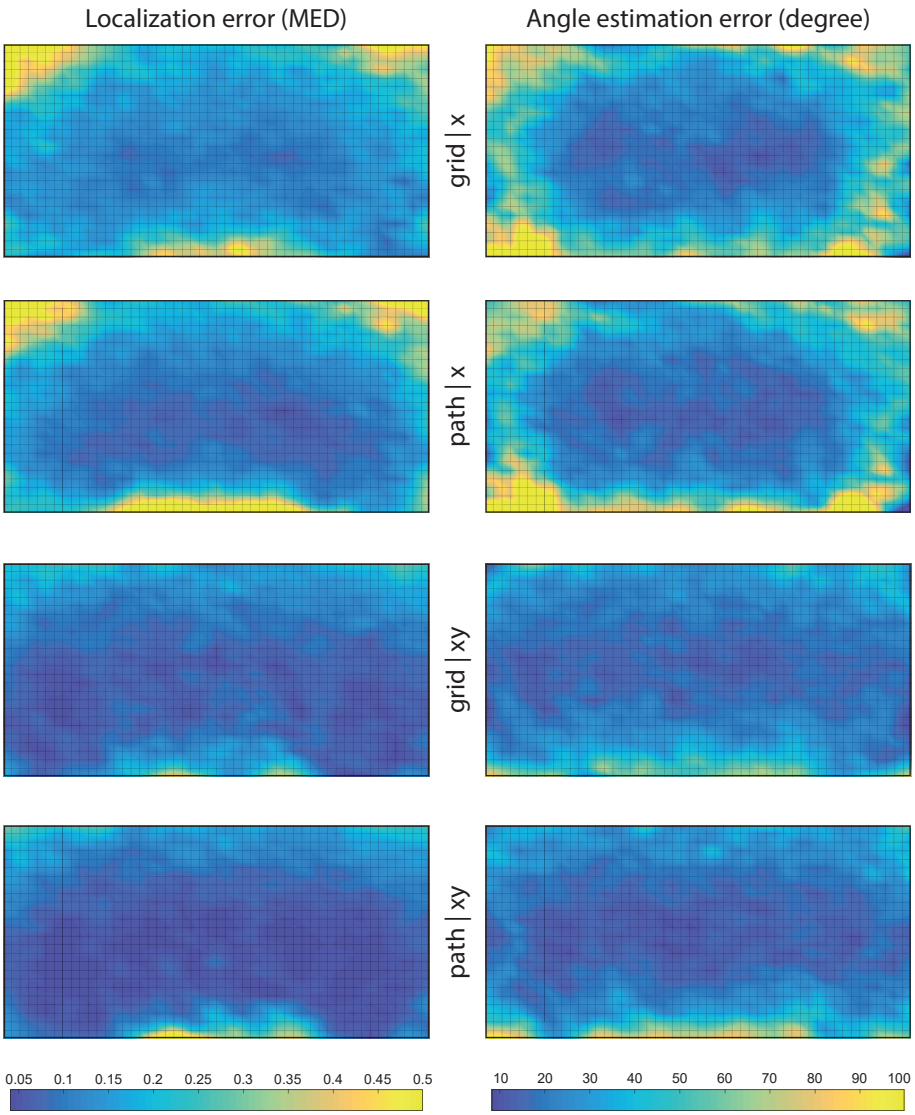


Figure 5.6: Averaged (N = 5) performance on path test set indicated on the $2L \times L$ area of interest ($\eta = 10^{-6} L$ m/s). The sensors are placed under the 'bottom' of the area of interest, as indicated in Figure 5.2.

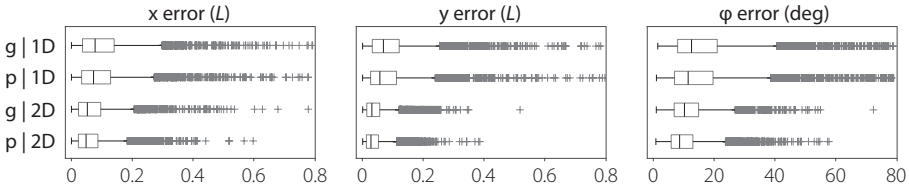


Figure 5.7: Box plots of separate estimation errors ($\eta = 10^{-6}L$ m/s).

For the location error at high noise levels, each case has a different plateau level. All methods outperform random chance, which for a 2×1 area amounts to 0.805 [85]. For the orientation estimation error at high noise levels, each case plateaus at 90 degrees which is at chance level, as it is half of the maximal error of 180 degrees.

5.5 DISCUSSION

5.5.1 *ELM performance*

Regardless of strategy and sensor type, the ELM neural network architecture is shown to be capable of determining both the location and orientation of a moving submerged object using an array of flow sensors.

The ELM outputs are learned from the hidden layer representation independently by design; the architecture therefore prevents making use of a known time-relation between location and orientation. When a source is moving in a path, the current location and orientation very well describe the next location of a source; therefore, taking past velocity profiles or estimations into account might also improve results. Other (neural network) algorithms might therefore be more suited for artificial lateral line source detection in a path setting. However, echo state networks and MLPs have been outperformed by ELMs for this path setting elsewhere [20].

Combining two sub tasks might have slightly impaired the estimation performance. As is indicated by the validation error during the hyper-parameter optimization (figure 5.5), the optimal hidden layer size is higher for determining the location compared to determining the orientation. The chosen hidden layer size for ELM may have caused the network to slightly under fit on the localization task while being slightly over defined for determining the orientation. Separating these tasks to different networks in future work may therefore improve the results of both sub tasks.

5.5.2 *Data set strategies*

On average, the path strategy outperforms the grid strategy. Especially in the center areas of Figure 5.6, path shows a lower error. This comes at the cost of performance

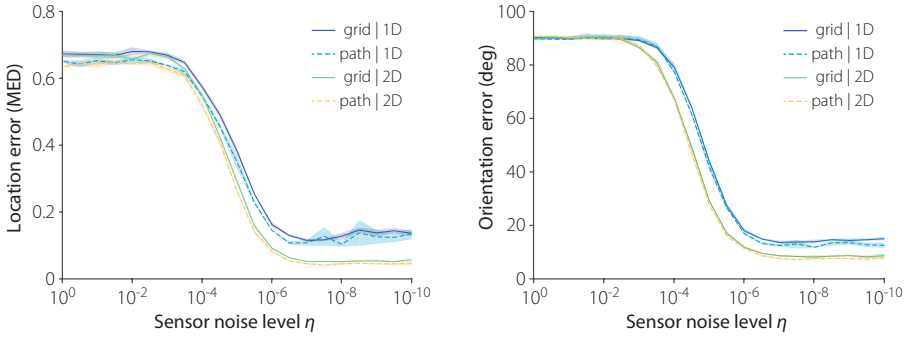


Figure 5.8: Averaged performance (lines) and standard deviation (fills) on varied sensor noise levels (N=5) for both type of sensors and data sampling strategies.

near the artificial lateral line. This is possibly due to the fact that the grid strategy trains the neural network at all locations, while a generated path is less likely to produce data near the array. This trade-off between estimation performance near a sensor array and further away could be taken into account when choosing a strategy.

5.5.3 *Sensor types*

As anticipated, the 2D sensors benefit the source detection performance for both location and orientation. While the extra inputs from the orthogonal velocity profile also help to reduce the effect of sensor noise, another effect is apparent in the orientation estimation error in Figure 5.6. In the bottom corners of the area of interest, the performance increases considerably. A possible explanation originates from the contribution of the source orientation in constructing a velocity profile. The profile is a mix of odd and even mother wavelets; its ratio depends on the source angle φ [31]. The lower corner areas are nearly in line with the sensor array and therefore produce weak sensor readings. However, with both velocity components, two perpendicular mixes (see Figure 5.2, right) of the mother wavelets are measured. Because determining the ratio of mother wavelets is more reliable with two perpendicular velocity profiles, the orientation estimation uncertainty and therefore error decreases.

5.5.4 *Influence of noise*

The noise magnitude η affects the maximal distance with which the system can track a source. The localization error at very low noise levels (Figure 5.8), shows an increase in error for the 1D sensors. This indicates that the neural network might start to over fit on the training data set. This may be remedied by increasing the number of training data

or reducing the hidden layer size. Regardless of this increase, the path strategy with 2D sensors performs best for a wide range of noise levels for localizing a moving source.

5.6 CONCLUSION

The ELM architecture was able to solve the inverse problem to a high degree; we were able to predict a source's location and orientation based on the discrete measurement points from a fluid flow sensor array. The lowest average location error was 4.2% ± 0.25% L and the lowest orientation error was 7.1 ± 0.23 degree, both from the optimal combination of the path method and using 2D sensors.

Both data set strategies have advantages. The grid strategy produces unbiased data, with an increase in performance near the sensor array. The path strategy results in a lower average estimation error in our simulation. This might be because the simulated motion produces more relevant location-orientation pairs in the data set for this task, while introducing a slight bias. For physical arrays, the path method might be the optimal choice. Large path data sets can be readily produced using a tracked source moving near an array, while grid requires multiple discrete measurements. Since larger path data sets can be made relatively quickly, this might compensate for the slight bias.

For both strategies, the 2D sensors, which are sensitive to two perpendicular directions of flow, outperform the 1D sensors, which are only sensitive to the fluid flow parallel to the array. This is likely due to the complementary role of the orthogonal v_y velocity profile.

Part III

LOCALIZATION EXPERIMENTS

The experiments described in this part of the thesis are based on the sensor described in part I, and expand on the simulation studies from part II.

Chapter 6 describes a small-scale experiment for localizing both a moving sphere and a vibrating dipole using an array of four sensors, further demonstrating the advantage of 2D sensing. In chapter 7, an array of eight sensors is used for localizing a moving sphere. Two types of artificial neural networks are compared to attest the relevance of sensing history and recurrency in neural networks for this localization task.

6

SMALL-SCALE EXPERIMENTS FOR DIPOLE AND MOVING OBJECT LOCALIZATION

As an initial demonstration of the techniques presented in part II, we present two small-scale experiments for localizing a vibrating dipole source and a moving object, using an array of four sensors.

We first extend a wavelet-based description for the generated fluid flow of sphere in motion to 2D. This allows describing the 2D velocity profile generated by a moving sphere, using three basis functions. By positioning a dipole source on several locations within a grid, we first investigate whether the measured velocity profiles match with the theoretical description. We then use an ELM neural network to localize the dipole source and report the localization error.

For the second experiment, we use a 2D plotter to move the object through the area of interest with constant speed. Since the objects position varies with time, we consider several preprocessing steps, including taking past velocity profiles into account. Via spatial analysis of the localization error, we assess the performance of the localization in both the average localization error and the detection range.

PROVENANCE This chapter has been previously published as:
B.J. Wolf, S.M. van Netten, (2019). Hydrodynamic imaging using an all-optical 2D artificial lateral line. *2019 IEEE Sensors Applications Symposium (SAS)* 1-6.

AUTHOR CONTRIBUTIONS BW conceived the moving object study, extended the theory, built the setup, performed the experiments, performed the analysis, wrote the draft, and wrote the final manuscript. SvN conceived the dipole study, supervised the study, and revised the draft.

ABSTRACT Fish and amphibians can sense their hydrodynamic environment via fluid flow sensing organs, called lateral lines. Using this lateral line they are able to detect disturbances in the hydrodynamic near field which enables hydrodynamic imaging, i.e. obstacle detection. Via two experiments we demonstrate a novel artificial lateral line of four bio-inspired 2D fluid flow sensors and show that the measurements of the enacted sensors agree with an established hydrodynamic model. These measurements from the array are then used to localize both vibrating and unidirectionally moving objects using an artificial neural network in a bounded area of 36 by 11 cm which extends beyond the area directly in front of the sensor array. In this area, the average Euclidean localization error is 1.3 cm for a vibrating object, while for moving a object it is on average 3.3 cm.

6.1 INTRODUCTION

The development of our sensory array is based on a biomimetic approach as the design is directly inspired by a sensory modality used for flow detection by aquatic vertebrates; fish and amphibians have the unique ability to detect and localize moving and vibrating underwater objects [36]. Along the head and trunk they have an array of discrete mechanical sensors at their disposal called neuromasts. With these neuromasts they can perceive the local water motion – or flow – relative to their body.

Artificial lateral lines (ALLs) provide a passive sensor system that works in the hydrodynamic near field which does not rely on vision or an active beacon. Other technologies, e.g. sonar, do require an active beacon which is detectable and can affect marine ecosystems.

ALLs can be used in two modes. It can be used to help autonomous underwater vehicles (AUVs) safely navigate murky waters in terms of obstacle detection, bridging the gap between touch and vision. Secondly, the system could be used in harbors and other waterways to passively monitor marine traffic or marine life.

Several implementations of ALLs exist [81]. Most implementations are scaled down versions called mechanical micro sensors (MEMs), which make use of piezo-strain sensing, while others use pressure sensors or optical levers with LEDs [58]. The present work demonstrates the first use of all-optical sensors in an ALL, which does not rely on electrical signals and could therefore be more reliable in operation. In addition, some ALLs [8, 127] show varying sensor orientations, allowing to partly capture the information of fluid flow perpendicular to the array. The presented sensor array is one of the first artificial lateral lines that combines the parallel and orthogonal velocity profile components in single measurement points, which has been shown to be beneficial for location estimation in simulation [122].

In this study, we demonstrate hydrodynamic imaging [125], i.e. the ability to detect an object near the sensor array, via two benchmarks. The first benchmark task sets out to localize a stationary vibrating (dipole) object positioned at discrete locations, while the second task involves tracking and localizing a unidirectionally moving object.

To estimate the location from the sensor data, we make use of a fast artificial neural network architecture. This type of neural network is also used in [20, 122], which are both simulation studies for continuous moving objects; whereas the current study aims to demonstrate hydrodynamic imaging via real-world experiments.

The rest is structured as follows: In section 6.2 we further describe the nature of velocity profiles, the sensors, and other relevant literature. In section 6.3 and 6.4, we present experiments for localizing a vibrating object and a moving object respectively. We discuss our findings in section 6.5 and conclude in section 6.6.

6.2 BACKGROUND

6.2.1 *Bioinspiration*

The fish lateral line is made up from two types of mechanosensory neuromasts, each with their own beneficial physical properties to help the fish perceive freestream (DC) and dynamic or oscillatory (AC) flow [36]. Superficial neuromasts (SNs) are present on the fish bodies' surface and are in direct contact with the surrounding. They are tailored to perceive steady flow i.e. fluid velocity. Canal neuromasts (CNs) are embedded in subdermal canals and therefore shielded from the DC flow and tailored towards AC fluid flow.

The perceived local fluid flow can be concatenated to a spatial velocity profile through combining information from other CNs and SNs in the lateral line system, which augments the fish sensory perception [28]. In behavioral fish experiments, the lateral line has been shown to be instrumental in many specific behaviors, for instance: prey detection, predator avoidance, schooling behavior, and spatial orientation [28]. The sensors described in this work are used as superficial neuromasts.

6.2.2 *Velocity profiles*

In previous research, theoretical models are described that predict velocity profiles (i.e. the local fluid velocities at each sensor unit on a given time) along a 1D array. Here, an object is either vibrating in a direction with a specific angle with respect to the array [6, 31, 33, 50] or moving in a specific direction [43]. Using these models based on non-viscous flow around a sphere, velocity profiles can be modeled for different locations and movement directions for a spherical object located at coordinate (b, d).

The local fluid velocity component parallel to the array, v_x, on position s on the sensor array is given by a combination of two wavelets [31]. The magnitude of the velocity

profile scales with the radius of a sphere a, its speed magnitude W, and its distance d to the array.

$$v_x = \frac{Wa^3}{2d^3}(\Psi_o \sin\varphi - \Psi_e \cos\varphi). \tag{6.1}$$

The angle of the object with respect to the sensor array is φ. The even wavelet Ψ_e and odd wavelet Ψ_o are described by:

$$\Psi_e = \frac{1-2\rho^2}{(1+\rho^2)^{\frac{5}{2}}}, \tag{6.2}$$

$$\Psi_o = \frac{-3\rho}{(1+\rho^2)^{\frac{5}{2}}}, \tag{6.3}$$

$$\rho = \frac{s-b}{d}, \tag{6.4}$$

where $-\Psi_e$ describes the v_x profile when the object is moving parallel to the array and Ψ_o when the object is moving orthogonal to the array ($\varphi = \pi/2$).

We extend this family of wavelets to also consider the velocity profile component orthogonal to the array, v_y. Here, Ψ_o describes v_y when the object is moving parallel to the array and Ψ_n when it is moving orthogonal to the array:

$$v_y = \frac{Wa^3}{2d^3}(\Psi_n \sin\varphi + \Psi_o \cos\varphi), \tag{6.5}$$

$$\Psi_n = \frac{2-\rho^2}{(1+\rho^2)^{\frac{5}{2}}}. \tag{6.6}$$

A graphical representation of the wavelet family and velocity profiles can be found in figure 6.1. At increasing distances, the velocity profiles decrease in magnitude, but also undergo spatial broadening [31]. The broadening is reflected in the minima, maxima, and zero crossings of the velocity profile; they displace further from the center of the profile, while the general shape remains.

6.2.3 *Optical 2D sensing*

Our sensor array consists of four novel, isotropic, all-optical, 2D-sensitive fluid flow sensors [124], which enables measuring a planar projection of the hydrodynamic environment. These sensors consist of a fluid force recipient sphere and a fiber support, constructed from four standard communication SMF-28 fibers.

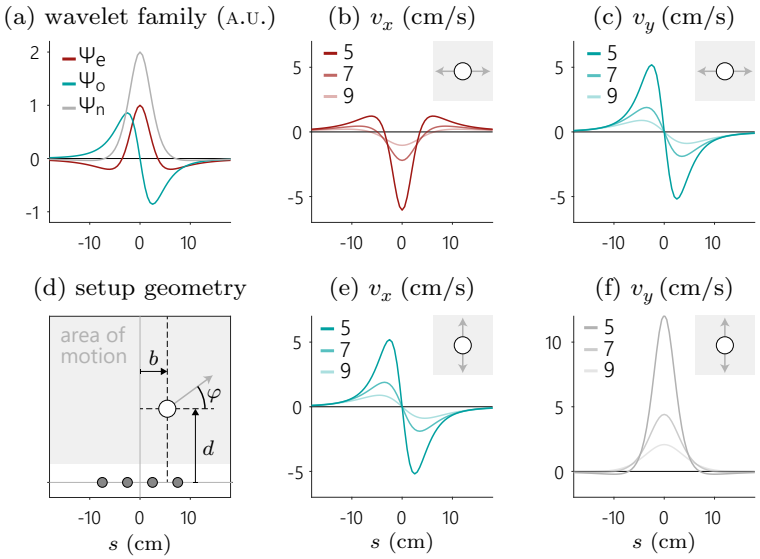


Figure 6.1: Depiction of the wavelet family (a), the setup geometry (d), and example velocity profiles. The setup geometry (d) depicts the location of the four sensors (grey spheres) and an object (white) moving in a certain direction φ with respect to the array at coordinate (b, d). subfigures (b, c, e, f) show the expected velocity pattern for a vibrating object at 5, 7, and 9 cm distance.

The four fibers are each inscribed with a fiber Bragg grating (FBG) near the sensor base and have an associated distinct Bragg wavelength. The Bragg wavelength denotes a reflectance peak, which can be measured via an optical interrogator. When the sensor is deflected, the internal fibers stretch and compress, producing a measurable Bragg wavelength shift, see also figure 6.2. The FBGs are also linearly affected by temperature, but this effect is smaller and can be factored out by using the wavelength shift difference between pairs of FBGs, since the temperature difference between FBGs in a sensor unit is negligible [39].

In chapter 3 we show that this sensor design is isotropic, i.e. equally sensitive in both directions [124]. Furthermore, a hydrodynamic model inspired by the fish neuromast provides an analytical frequency response that can be used to relate sensor deflection to local fluid velocity, taking into account both inertial and viscous drag fluid forces.

6.2.4 *Processing methods*

For both simulated and physical artificial lateral line arrays, several algorithms have been put forth to decode a velocity profile to retrieve the location of a dipole source. In

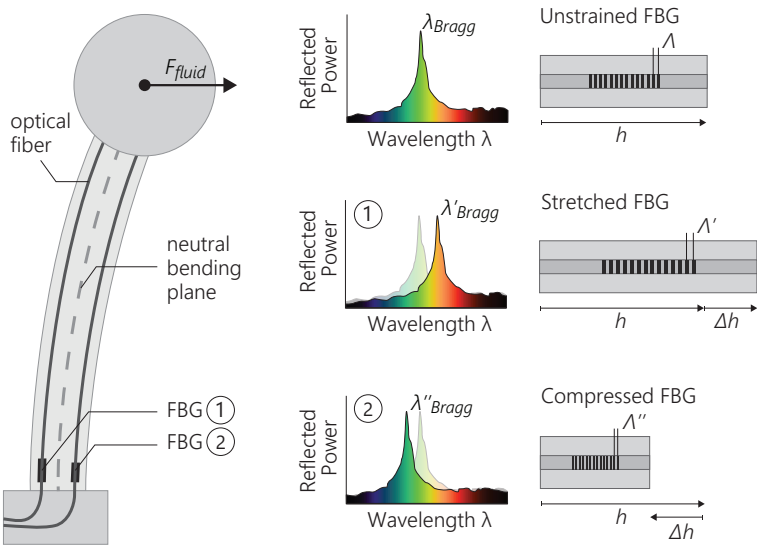


Figure 6.2: Schematic overview of sensor design and sensing principles, adapted from [124]. When the sensor is deflected, opposite fiber cores will stretch and compress respectively, producing a shift in the measured Bragg wavelength.

[31], Curcic-Blake and van Netten proposed a continuous wavelet transform, based on the mother wavelets as described in section 6.2.2. A data-matching approach where a measured excitation pattern is compared to a large set of templates was suggested by Pandya et al. [96]. This template matching approach was later shown to be outperformed by Capon's beamforming algorithm [127]. More recently, artificial neural network architectures, including the Extreme Learning Machine, have been used to determine an object's location [4, 20] and orientation [122].

6.2.5 *Neural network*

To demonstrate the localization performance on a limited size data set with our array, we make use of an efficient single-layer feed-forward neural network, the Extreme Learning Machine (ELM) [64]. This type of neural network has a single hyper parameter, the size of the hidden layer, which allows for fast optimization while preventing over fitting. This type of neural network has shown the best performance for hydrodynamic object localization compared to more advanced neural networks [20].

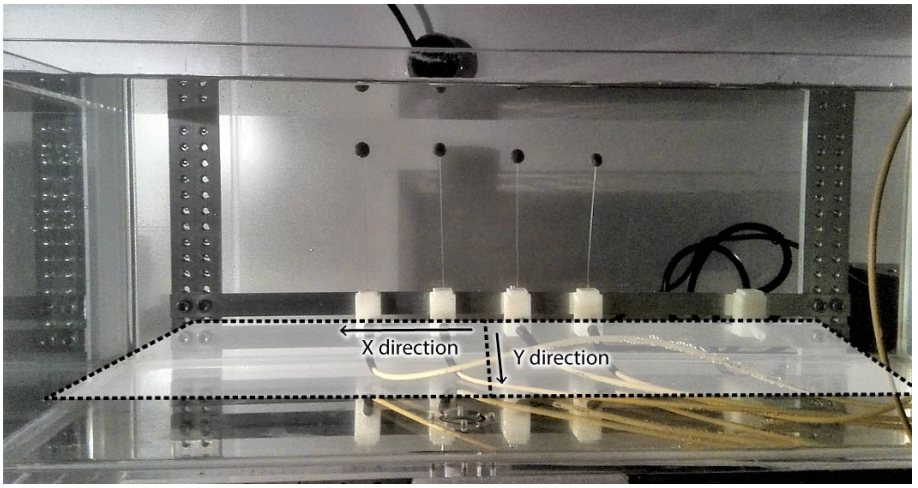


Figure 6.3: Four sensors are mounted 3 cm from the back wall of a 25 × 50 cm testing tank. The stimuli are leveled with the fluid force recipient spheres during the experiments.

6.3 DIPOLE OBJECT LOCALIZATION

In this experiment, we benchmark hydrodynamic imaging by the task of localizing a dipole source (i.e. vibrating object) positioned at discrete locations.

6.3.1 *Setup*

In our experimental setup, four sensors form an array of 14 cm, or artificial lateral line, at one side of a water tank measuring 25 × 50 cm, see figure 6.3. The sensors are positioned 3 cm from the back wall via a guiding rail parallel to the long side of the tank. The drag force recipient spheres are leveled about 3 cm under the water surface. During all experiments, the object is leveled with the sensors. The Bragg wavelengths of the sensors are measured at 1 kHz using an optical interrogator (Micron Optics sm125, Atlanta, USA).

6.3.2 *Methods*

A B&K 4810 mini shaker is used in combination with a signal generator to vibrate the submerged sphere, 6 cm in diameter, at 8 Hz with a motion amplitude of 1 mm. We direct the motion towards the array ($\varphi = -\pi/2$), we therefore expect to see Ψ_o in the x-component of the measured velocity pattern and Ψ_n in the y-component (see section 6.2.2).

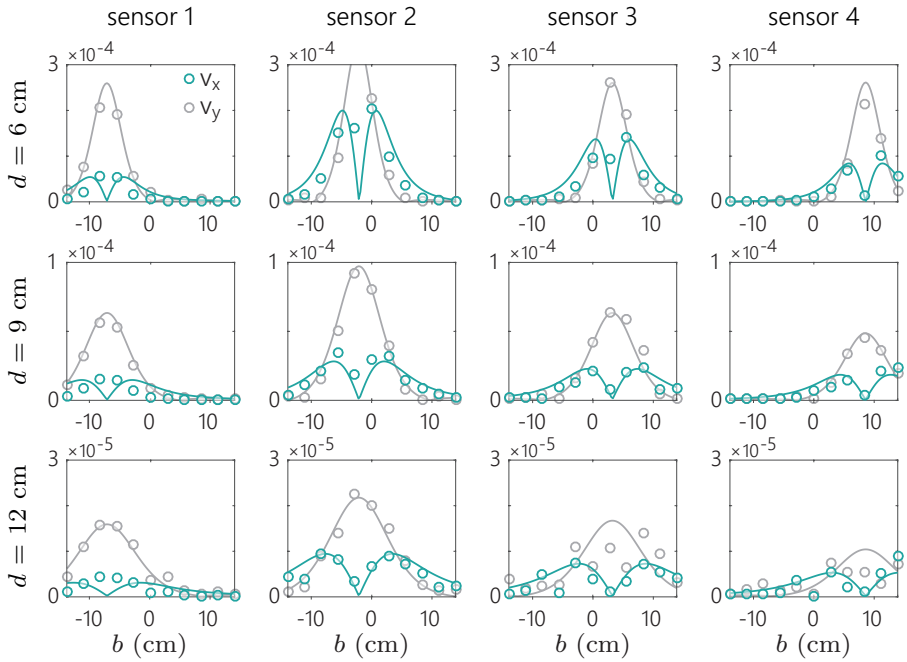


Figure 6.4: For each sensor, the response magnitude (m/s) for 11 lateral positions b per distance d is shown. Note that for increasing distances, the y-axes are scaled as to make the spatial properties visible for decreasing magnitudes.

For each location, we record 40 seconds of sensor data while the object is vibrating. Via a discrete Fourier transform, we obtain the magnitude of fluid flow at the stimulation frequency for the four sensors and 2 dimensions, reducing the sensor data to a vector of 8 points per location.

6.3.3 *Results*

Since the current array lacks the spatial resolution to properly assert the velocity profile, we first present the measured flow velocities per sensor rather than location. In figure 6.4 we show the measured amplitudes for 11 lateral locations at three distance levels together with a parametric fit based on the expected wavelets. Especially for the profiles measured close to the array, the sensor readings correlate very well with theory. At further distances, both the effects of reduced amplitude and spatial broadening of the velocity patterns are present.

Finally, we trained an ELM neural network for localizing the object based on the sensor responses, the results of which are shown in figure 6.5. Here, we used a 5-fold

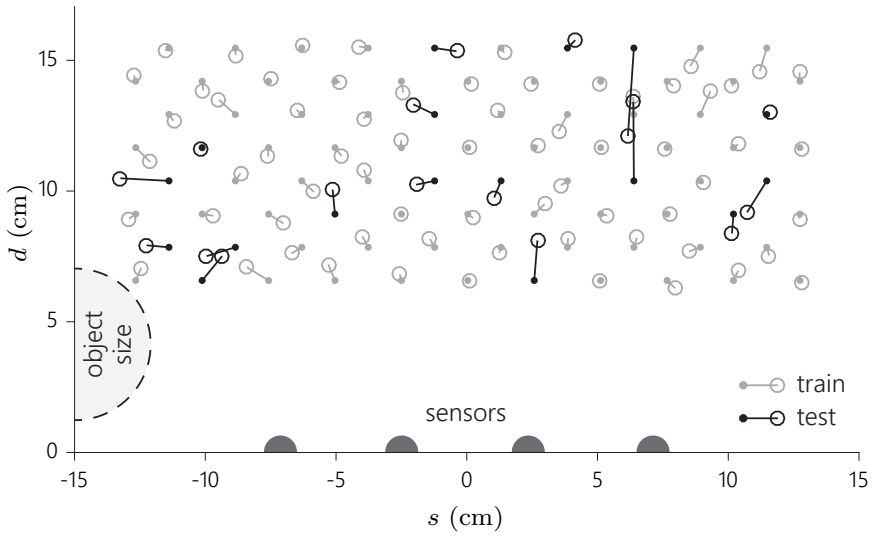


Figure 6.5: Localization results for a single fold of the dipole localization benchmark. The estimated locations '○' are linked to their respective true location '●'. The distance error of the training set (gray) is slightly lower than the unseen test set (black).

validation on 84 locations, where a random 80% portion of the measured locations is used for training while the remaining 20% is given to the neural network to predict. Therefore not all locations are used for training. The ELM hidden layer size was 1000.

We calculate the mean Euclidean localization error by combining the 5 folds and averaging the error distances of the 5 test sets and the 5 training sets. This error for the (unseen) test set was 1.3 cm, while the (seen) training set had a better estimate with an error of 0.45 cm. The ALL and ELM are therefore able to reconstruct locations from excitation patterns for unseen locations (figure 6.5).

6.4 MOVING OBJECT LOCALIZATION

As was shown by Franosch et al. [43], the velocity patterns resulting from motion with constant speed are, ignoring vorticity and other viscous effects, identical to those from dipole sources. However, the object is moving rather than vibrating, we therefore adapt the setup and apply different preprocessing steps before feeding the data to the ELM neural network for localizing the object.

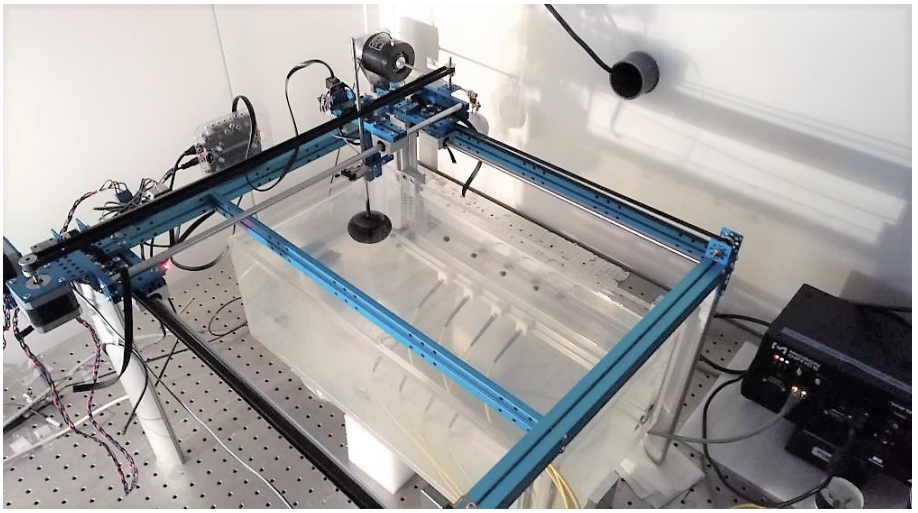


Figure 6.6: Adapted setup for moving the object with the 2D plotter (blue). The optical interrogator is visible in the bottom right corner.

6.4.1 *Setup*

For this bench marking task, the water tank setup is fitted with an Arduino controlled XY plotter (Makeblock) to enable an object to be moved in a straight path at constant height. Both the optical interrogator, which reads out the sensors, and the Arduino controller board are controlled through a Matlab script to allow for simultaneous stimulation and recording of sensor data. Furthermore, the Arduino board reports its location on preset intervals as to provide accurate training labels, i.e. annotated locations.

The object (6 cm diameter) is leveled with the sensors and moved at a constant speed of 7 cm/s, with a ramp up and down at the start and end of its motion. The sensor positions have been changed to form a slightly smaller array of 12 cm.

6.4.2 *Preprocessing and validation*

To determine valid preprocessing steps and the hyper parameter of the neural network, we first record a pilot data set of motion parallel to the array at six distances from 4 to 9 cm. This pilot data set consists of 10 repetitions of both forward and backward motion for each distance, so 120 runs in total. We use 5-fold stratified cross validation to validate the performance of the chosen settings.

We varied the hidden layer size between {1000, 1500, 2000, 2500}. Furthermore we varied a down sampling factor between 1 (no down sampling) to 53. Since this experiment produces time traces, we can take past detection into account. We implement this via

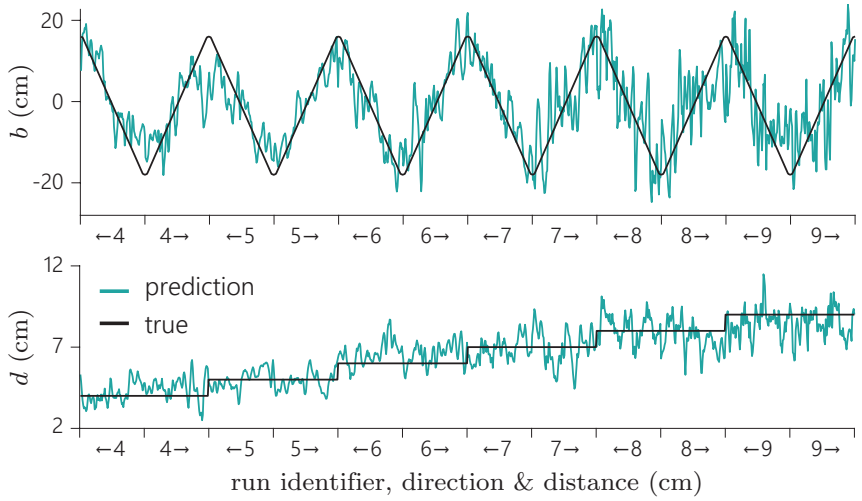


Figure 6.7: Prediction and true location of the object for the pilot data test set for one fold. Shown here is the concatenation of 12 different run cases.

feeding time windows to the ELM, rather than a single time step. To determine the optimal window length, we vary the history window size from 1 (no history) to 45. To make sure that the algorithm can provide timely output, we limit the amount of effective history (Δt per frame after down sampling × history size) to 500 ms.

From these settings, the estimator with the lowest test error has a hidden layer size of 1500, down samples by averaging 37 samples and takes 13 history steps as input and therefore requires a history buffer of 481 ms.

The output on the test set of a single fold of the optimal estimator is shown in figure 6.7. Here we observe that the prediction of the x-coordinate (b) is of better quality than the y-coordinate (d) prediction, which can be caused by the former having more variation in the pilot data set. Another observation is that the prediction performance decreases over the distance d, which aligns with the fact that velocity patterns from objects at distances further away also have a lower signal to noise ratio.

6.4.3 *Methods*

For the final benchmark experiment, the XY plotter is instructed to move the object back and forth 5 times across 313 straight paths within an x,y bounding box of 36 × 11 cm. Figure 6.8 indicates this bounding box and shows a small selection of the used paths. Here, b was divided in 13 points from -18 to 18 cm, and d was divided in 12 points from 4 to 15 cm.

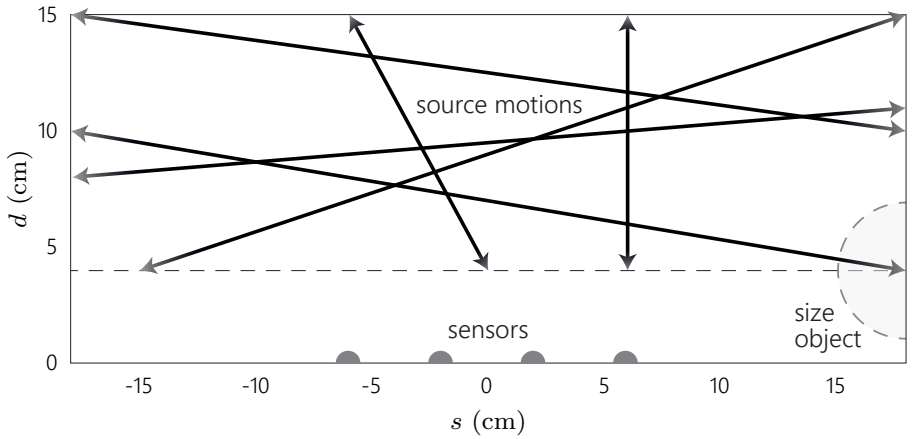


Figure 6.8: A selection of the paths in the data set in a 36 × 11 cm bounded area. All motions are straight and edge-to-edge within the bounding box. The gradient on the motion arrows resemble the ramp up and down.

The set of motions was constructed from all points of one side to all points on its opposing side. The total data set consists of 3130 runs. With this set, we apply 5-fold stratified cross validation (four of the recorded sets of motions are used for training; the fifth set is used for testing).

6.4.4 *Results*

The averaged Euclidean error on the whole test set is 3.3 cm (see also figure 6.9) and 3.2 cm for training. The localization error is comparable to the object radius and inter-sensor distance, but other relative interpretations are possible. Furthermore, the very small difference between the training and testing error indicates that there is very little over fitting taking place.

These results indicate that moving object localization is feasible with an all-optical ALL of four sensors.

6.5 DISCUSSION

We first showed that the 2D velocity profiles as measured by the four-sensor artificial lateral line (ALL) agree with those derived from a hydrodynamic model. Even though the sensors are not equally sensitive, the spatial variations such as wavelet broadening, zero crossings and maxima are clearly present in the measured data. Based on the sensitivity

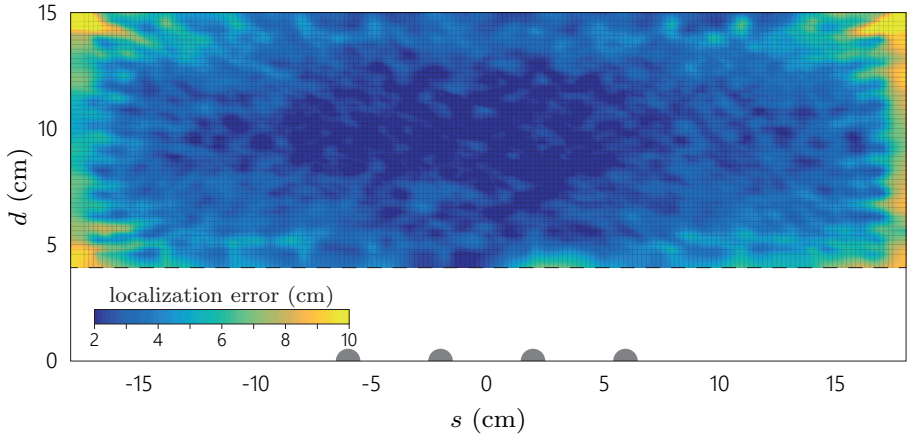


Figure 6.9: Meshed interpolation of the Euclidean localization error with respect to the true location of the object. While the edges display an increase in estimation error, a large central area in front of the array has an error comparable to the object radius and could be interpreted as correct localization.

of the individual sensors, these spatial variations are progressively obscured at further distances, as expected.

The first benchmark test for hydrodynamic imaging was to localize a vibrating sphere. Using a small data set (< 100 samples) of discrete object locations, we were able to show that an extreme learning machine (ELM) neural network can be easily trained to reliably predict an object's location based on the amplitudes of the excitation profiles. The mean Euclidean prediction error on the test set with 67 trained locations per fold was 1.3 cm in a bounded area of 27 × 11 cm. In a different dipole localization task [4], they achieved an average localization error of 0.28 cm with 90 training points and 4 enabled sensors in a 18 × 9 cm bounded area with a stronger source. With 50 training points, they achieved an average error of 0.4 cm. Between our area being larger, using a lower intensity dipole source, and the vibration direction being different, it is difficult to conclude which algorithm or array performs best based on the current experiment.

From the ELM and preprocessing optimization for moving objects, we found that considering past perceived excitation patterns helps in increasing localization performance. The most influential factor was not the amount of past inputs, but rather the effective amount of history taken into account. This makes sense from the point of view that excitation patterns not only encode location, but also direction: the object's direction remains the same throughout a path of motion. Data sets with other, non-straight motions might therefore call for taking a shorter amount of history into account.

The final performance on the second benchmark test came to an average Euclidean localization error of 3.3 cm in a bounded area of 36 × 11 cm. This task is somewhat

harder to achieve, since there is no direct mapping from location to perceived excitation pattern; the object passed some points several times, but from different angles. To the best of our knowledge, there are no direct experimental comparisons for this task in the literature. Simulation results with 16 sensors and an ELM [122] indicate that an error of 0.9 cm is the lower bound for a 12 cm array in noiseless conditions. Considering that our error of 3.3 cm is in the same order of magnitude, it performs quite well.

The current sensors have shown to be sufficiently sensitive for the hydrodynamic stimuli used in this study, but tied with their sensitive nature is that the sensors are fragile. In the current environment we controlled the object's position, but for future deployments these sensors could be fitted in protective housings, which may also increase the robustness to noise and DC offsets for dynamical signals [128].

Another step to improve the accuracy concerns the number of sensors and the array geometry. As shown in [20], the number of sensors greatly affects localization performance; a first step would be to increase the spatial resolution of the array. Here, the inter-sensor distance can also have an influence [6]. Other sensor configurations than the bio inspired equally spaced line might also improve the performance.

Furthermore, given that the predicted location for a moving object is quite noisy (e.g. figure 6.7), post processing methods such as particle filters or median filtering could be considered for stabilizing the network output.

6.6 CONCLUSION

Both benchmark tests were successful in the sense that object localization was possible using the novel, all-optical, 2D artificial lateral line (ALL) whether the object was vibrating at discrete positions or unidirectionally moving through a bounded area. We consider this result encouraging, even more so when considering that these results only stem from a relatively sparse four-sensor artificial lateral line.

Summarizing, our results show that the current sensor design and ALL implementation is very well suited for hydrodynamic imaging of moving and vibrating objects. The current sensor design, results, and analysis provide promising future developments.

7

MEDIUM-SCALE EXPERIMENTS FOR MOVING OBJECT LOCALIZATION

This chapter features a demonstration on a larger scale compared to chapter 6. An array of eight sensors placed in a 120 × 80 cm tank is used to measure the 2D velocity profiles resulting from an object moving near the sensor array.

In this study, we investigate whether recurrent connections are advantageous for object localization. In chapter 3, we found that in some cases, the echo state network outperformed the other feedforward neural networks in simulation. We therefore compare the performance of a recurrent neural network (LSTM) with an feedforward neural network (OS-ELM), to see whether this finding also holds in for experiments using 2D-sensitive sensors.

We assert the localization performance of both neural network types and their robustness with respect to signal to noise levels via increased noise on the inputs. Both the spatial analysis of the localization error and the effect of the noise levels are used to assert whether recurrency is beneficial for hydrodynamic imaging.

PROVENANCE This chapter has been previously published as:
B.J. Wolf, S. Warmelink, S.M. van Netten, (2019). Recurrent Neural Networks for Hydrodynamic Imaging using a 2D-sensitive Artificial Lateral Line. *Bioinspiration & Biomimetics* 14(5) 055001.

AUTHOR CONTRIBUTIONS BW conceived the study, built the setup, supervised the experiments, performed the final analysis, wrote the draft, and wrote the final manuscript. SW performed the experiments, performed the initial analysis, and revised the draft. SvN supervised the study and revised the draft.

ABSTRACT The lateral line is a mechanosensory organ found in fish and amphibians that allows them to sense and act on their near-field hydrodynamic environment. We present a 2D-sensitive Artificial lateral line (ALL) comprising eight all-optical flow sensors, which we use to measure hydrodynamic velocity profiles along the sensor array in response to a moving object in its vicinity. We then use the measured velocity profiles to reconstruct the objects location, via two types of neural networks: feed-forward and recurrent. Several implementations of feed-forward neural networks for ALL source localization exist, while recurrent neural networks may be more appropriate for this task. The performance of a recurrent neural network (the Long Short-Term Memory, LSTM) is compared to that of a feed-forward neural network (the Online-Sequential Extreme Learning Machine, OS-ELM) via localizing a 6 cm sphere moving at 13 cm/s. Results show that, in a 62 cm × 9.5 cm area of interest, the LSTM outperforms the OS-ELM with an average localization error of 0.72 cm compared to 4.27 cm respectively. Furthermore, the recurrent network is relatively less affected by noise, indicating that recurrent connections can be beneficial for hydrodynamic object localization.

7.1 INTRODUCTION

The lateral line is a near-field mechanosensory organ found in fish and aquatic amphibians. By measuring water displacement, these animals can detect objects in their vicinity without having to rely on vision or sound [36]. This allows them to perceive predators, peers and prey, even in murky or dark waters. This ability can be referred to as hydrodynamic imaging [30, 125] and can be discerned into two modes: *active* and *passive*.

In *active* hydrodynamic imaging, a fish generates a small flow field around themselves through movement. By detecting and measuring distortions in this self-generated flow field, they can detect stationary obstacles. Possible use cases include safe AUV navigation and obstacle avoidance [76, 117]. In contrast, *passive* hydrodynamic imaging relies on an external flow field or turbulences. These are generated by other objects that either move or are placed upstream in free stream flow, e.g. [22, 132]. This type of sensing could have applications in harbor security or tracking marine life without relying on an active beacon or vision in dark or murky environments.

Most biomimetic implementations of the lateral line, coined artificial lateral lines (ALLs), adhere to passive sensing [67, 81]; usually, an array of pressure or fluid flow sensors sample the hydrodynamic environment at discrete points which concatenate to spatio-temporal velocity profiles [31, 43]

To determine the performance of the ALL and signal processing pipeline, the combined system is usually bench marked via localizing a vibrating (dipole), or a moving object, or in one case a moving dipole [6].

Several signal processing methods have been put forth to use measured velocity profiles to recover the relative location of the object; thereby partly solving the inverse

problem [5]. These methods include Capon's beamforming [34, 95] and template matching [31, 96].

Since the relation between a source location and the resulting sensor signals is quite complex, Abdulsadda and Tan [3] and Wolf and van Netten [121] opted to use artificial feed-forward neural networks to perform this source localization task. The resulting systems were able to accurately detect the location of an object within a region of interest near the sensor array. In a comparison study [20], a recurrent neural network type (ESN) was shown to outperform feedforward neural networks in noisy conditions. This is likely due to the temporal correlation between subsequent object locations and resulting velocity profiles, i.e. temporal context.

In this research, we demonstrate passive hydrodynamic imaging, i.e. localizing a moving object in the vicinity of an artificial lateral line. We compare the performance of two neural network types, feed-forward and recurrent. We hypothesize that recurrent connections may help in localizing moving objects, since these networks can inherently make use of temporal context.

7.2 BACKGROUND

In this section, we discuss the biological lateral line and our bio-mimetic implementation, as well as the neural network architectures that support our experiments.

7.2.1 *Lateral line sensing*

The lateral line allows fish to perceive a plethora of hydrodynamic phenomena, ranging from movements as small as the water disturbance generated by plankton, to movements as large as river streams [30]. This phenomenon has been called *touch at a distance* [36], and has been associated with behaviors such as schooling, prey detection, rheotaxis, courtship, and station holding [30, 61, 89, 98, 105, 112]. Since lateral line perception does not rely on light or sound, it is still effective in environments where these are absent, such as in low-light (e.g. in a dark cave or in blind species) or low-visibility (e.g. murky waters) environments. Fish are capable of detecting sources up to a distance of roughly one fish's body length [26, 31, 69].

Lateral lines are comprised of distributed sensors called neuromasts. Two types of neuromasts exist: superficial neuromasts and canal neuromasts. Superficial neuromasts are found on the skin surface all over the fish body, and their number present in one individual ranges from very few up to thousands, depending on the species [27]. Canal neuromasts are found in fluid-filled canals under the outer skin. The canals themselves can usually be found along the trunk or near the head.

7.2.2 *Sensor array and operation*

Many different types of ALL sensors exist, although fluid flow is generally measured through deflection of a cantilever structure [81]. These sensors range from Microelectromechanical systems (MEMS) e.g. [13] and ionic polymer-metal composites (IMPC) [6] to optical guides [58] and, recently, all-optical 2D-sensitive deflection sensors [124].

The current all-optical sensor also senses fluid flow via a cantilever deflection. Each sensor contains four optical fibers with Fiber Bragg Grating (FBG) strain sensing elements. These FBGs reflect a specific wavelength of light, depending on periodic variations in the fiber core. When an individual fiber is strained, the periodic variation stretches or contracts, resulting in a measurable wavelength shift [42]. The fluid forces acting on the sensor are thus encoded in the sensor deflection, which can be measured as a linear function of change in reflected wavelength peaks [124].

7.2.3 *Neural networks for localization*

We discern between two different types of artificial neural networks: feed-forward networks and recurrent networks. The main difference between the two types of networks is the direction of the information flow.

In feed-forward networks, the input strictly moves forward, sequentially going through all layers until it reaches the output layer, which predicts the location of the source.

Conversely, recurrent neural networks have connections which allow the information to not only strictly go forward, but also allow connecting to the same layer, or even to a previous layer. Thanks to these recurrent connections, these networks can take history of the input into account, and are therefore often used when processing temporal data, such as time series.

7.2.3.1 *State of the art*

Artificial neural networks, such as the multi-layer perceptron (MLP) used by Abdulsadda and Tan [3] for locating a dipole source, are capable of learning complex, large, non-linear mappings from high-dimensional data sets.

Using simulated data, Boulogne et al. [20] compared the performance of an MLP, an echo state network (ESN) and the extreme learning machine (ELM) [63] on localizing a moving source under different levels of signal-to-noise ratio.

While the ELM outperformed the other algorithms in low noise conditions, the added complexity of the MLP and the recurrency of the ESN allowed these networks to take past velocity profiles into account, which may have caused this type of network to perform better with higher noise levels. This indicates that, under more noisy conditions (such as when using measured data), recurrent networks may be more suitable compared to feed-forward networks.

Specific to the all-optical sensors used in the current study, only the ELM architecture has been applied. Wolf and Netten [121] used data from an ALL consisting of 4 sensors, processed by an ELM with 1500 hidden units. The ELM used a time window of 481 ms as the input, which was down sampled to 27 Hz. The combined system was able to predict the location of a 6 cm diameter sphere moving at 7 cm/s with an average error of 3.3 cm in an area of 36×11 cm.

7.2.3.2 *ELM*

For our feed-forward neural network architecture, we make use of a variant of the efficient single layer ELM network. This type of neural network has a single tunable hyper parameter, the size of the hidden layer, which allows for fast optimization while preventing over fitting. The ELM's input-to-hidden weights α are randomized and fixed (figure 7.1a). This provides a hidden representation $\mathbf{H}$ and leaves only the hidden-to-output weights β to be learned [64]:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{7.1}$$

where $\mathbf{T}$ denotes the desired (teacher) outputs.

Using the Moore-Penrose generalized inverse $\mathbf{H}^{\dagger}$, the network is able to find optimal weights in a single learning step. An added benefit to the Moore-Penrose generalized inverse is that it produces the smallest norm solution, which also aids avoiding over fitting. This smallest norm, similar to L2-regularization [49], penalizes large weights and thus penalizes (over-)dependence on single input nodes.

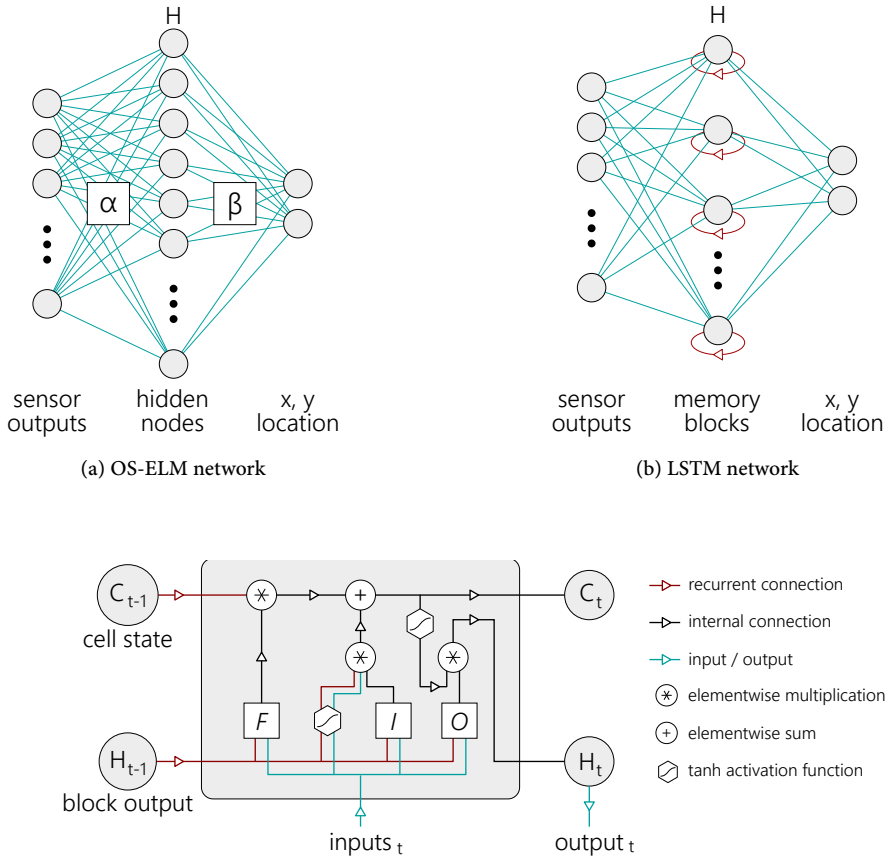
7.2.3.3 *OS-ELM*

One consequence of the single learning step, calculating the pseudo-inverse, is that this step becomes computationally intensive with larger data sets. Eventually, it is no longer computationally viable to perform the pseudo-inverse in one step due to hardware, i.e. working memory, constraints. The online sequential ELM (OS-ELM), which functions similarly to the ELM, can deal with larger data sets by implementing a sequential version of the least squares solution algorithm for the inverse operation.

Liang et al. [80] employ a sequential implementation by first calculating output weights based on an initial, sufficiently large, input batch. This initial batch produces a hidden activation or representation $\mathbf{H}_0$; consequently, its optimal output weights are calculated using the left pseudo-inverse via:

$$\hat{\beta}_0 = (\mathbf{H}_0^T\mathbf{H}_0)^{-1}\mathbf{H}_0^T\mathbf{T}_0, \tag{7.2}$$

This produces an initial weights estimate β_0, optimized for this initial subset. Subsequently, the network is given new input samples in small batches, and the output weight



(a) OS-ELM network

(b) LSTM network

(c) LSTM memory block

Figure 7.1: Overview of both neural network structures used in this chapter. Subfigures (a) and (b) show how the sensor output is propagated to the hidden layer via a set of input weights α. For the OS-ELM, each hidden node sums their input to a hidden value; all hidden values combine to vector H. For the LSTM (b), recurrent connections (red) make H dependent on time. Subfigure (c) further describes the inner mechanics of the memory block. The three gates (F, I, O) receive the past block output and current block inputs and each contain non-trivial processes. Via multiple operations, the cell state is updated and produces an output.

vector is updated based on the discrepancy between the target values of the next subset $\mathbf{T}_1$ and the estimate based on the previously found weights $\mathbf{H}_1\beta_0$:

$$\beta_1 = \beta_0 + (\mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T(\mathbf{T}_1 - \mathbf{H}_1\beta_0). \tag{7.3}$$

This second step repeats until all training samples have been processed. In the special case where the initial batch size equals the number of training examples, the OS-ELM implements the original ELM [80].

7.2.3.4 *LSTM*

The Long Short-Term Memory (LSTM) network is a recurrent neural network (RNN) capable of learning long-term dependencies. Invented by Hochreiter and Schmidhuber, the LSTM has changed very little since its conception in 1997.

Recurrent neural networks are networks in which nodes may have a connection to themselves. This recurrent connection allows them to take earlier inputs into account, giving these networks a temporal dimension. RNNs can therefore detect relations between events which are separated in time [99].

LSTM networks can learn dependencies between samples with a time lag of over 1000 samples [60]. This property has made LSTM networks a popular choice for problems in which long-term memory is needed, such as protein structure prediction [109] and speech recognition [52].

LSTM networks consist of complex nodes called memory blocks, see figures 7.1b and 7.1c. Each memory block contains a memory cell, whose state is affected by three gates: forget, input, and output. These gates roughly correspond to resetting, writing, and reading operations on this memory cell [51].

During the training phase of the network, the gates within the LSTM block are optimized via back propagating the prediction error through time. The steps for updating the input, recurrent, and bias weights as well as the update steps of each type of gate are omitted here and thoroughly described for several variations of the LSTM by Greff et al. [53].

7.3 METHODS

Here, we first describe the setup including the stimulus and sensor array. Then, we discuss the preprocessing steps and how we determine optimal parameters for the neural network implementations and preprocessing methods.

7.3.1 *Setup*

Data is gathered in a 1200×800×260 mm (w×l×h) water tank. A sphere with a diameter of 6 cm is moved horizontally in this tank by an adapted XY plotter (Makeblock) at a velocity of 13 cm/s to create hydrodynamic stimuli. The plotter is controlled with an Arduino-compatible mainboard, which reports its location on set intervals.

The sensor array (figure 7.2b) consists of eight sensors with an inter-sensor distance of 64 mm. One of the eight sensors malfunctioned (see figure 7.3b), and its data is therefore excluded from further analysis, functionally making it a 7-sensor ALL.

Each sensor is protected by a perforated 32 mm plastic tube as a failsafe to avoid impacts with the object. The perforation pattern is chosen to be uniform with a ring of eight equidistant 9 mm diameter holes, sandwiched between two rings of eight equidistant 3 mm holes. This allows the bulk of the flow field to enact the sensor, while still providing structural integrity for protection. These perforated tubes may slightly affect the sensor sensitivity [124] and introduce a dampening effect; lowering the resonance peak magnitude and frequency, and reducing overall sensitivity.

The measured sensor deflections are obtained via an optical sensing interrogator (Micron Optics si255, Atlanta, USA) at a sampling frequency of 250 Hz. Each sensor is calibrated via a calibration procedure described by Wolf et al. [124]. Given that the sensors were calibrated for fluid flow speed outside of their protective tubes, we opt to use the measured deflections as inputs for the neural networks. The expected measured velocity profiles are low frequency (< 1 Hz) stimuli, we may therefore assume that the relation between the deflections and fluid velocity is close to linear [124].

Both the XY plotter mainbord and the optical sensing interrogator are controlled via a Matlab R2016a script. This allows synchronizing the sensor data with the stimulus location, providing accurate teacher labels for the neural networks.

7.3.2 *Data generation*

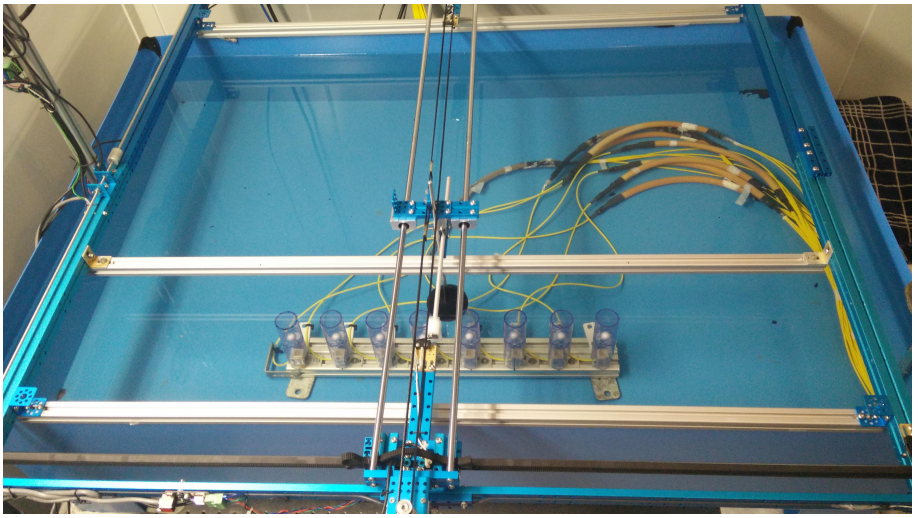
Object movement is constrained to the (2D) xy-plane; its depth is kept constant during all experiments. The sphere is moved along 256 unique linear paths in a region of interest in front of the array, see figure 7.2a. For each path, the sphere is moved back and forth 6 times between the left and right bound, creating a total of 3072 runs.

In addition to the recorded sensor signal, the plotter also provides timestamps for movement initiation, ramp-up, ramp-down, and movement end. These timestamps are used to synchronize and extract the parts of the signal where the velocity is constant; i.e., in between ramp-up and ramp-down. We combine the timestamps with the coordinates given to the plotter to automatically generate location labels for these segments with a constant velocity.

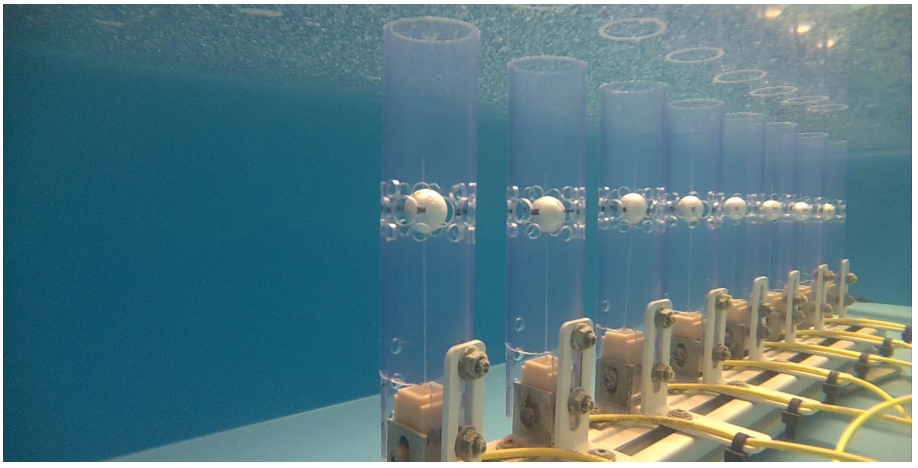
After each plotter movement, the plotter pauses for 20 seconds in order to minimize the effect of the previous movement on the hydrodynamic situation during subsequent movements.

7.3.2.1 *Preprocessing*

For preprocessing, we center the sensor x- and y-deflections around zero by subtracting their median from the signal. We also investigate the effect of down sampling the signal,

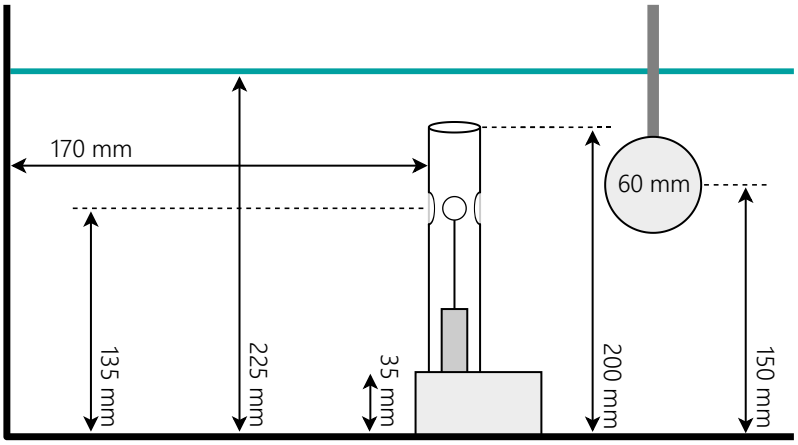


(a) Water tank with 2D plotter

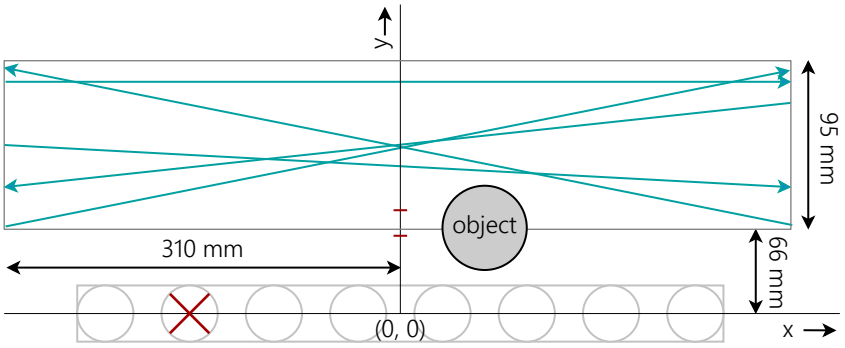


(b) Sensor array situated in water

Figure 7.2: Pictures of the setup. Subfigure (a) shows the 2D plotter (blue and blank aluminium beams) with respect to the sensor array. The horizontal blank aluminium bar above the ALL additionally protects the sensors from impact with the object. Subfigure (b) shows the 2D-sensitive artificial lateral line (sensor array) situated in the water tank. Eight sensors are placed equidistantly on a 45 cm guiding rail, with perforated protective tubing.



(a) Schematic side view of setup



(b) Schematic top view of area of interest

Figure 7.3: Schematic views of the setup. Subfigure (a) shows the dimensions and distances between fixed key points as used during the experiments. The distance between the object and the sensor array varied during the experiments. Subfigure (b) shows the area of interest, the object, motion paths, and the sensor array. The red cross indicates the excluded sensor. Teal arrows show examples of individual runs. The two red dashes on the y-axis indicate the two distances for the second type of experiments (see section 7.3.4.)

which can greatly decrease training time and serves as an additional form of noise reduction. Furthermore, it scales down the temporal dimension, making temporal correlation less distant and easier to learn.

Our parameter sweep for preprocessing includes three different down sampling ratios: 4×, 8×, and 16×.

The down sampled data set is subsequently normalized before feeding it to the neural networks. We test the performance using the original data and four variants of normalization techniques, all of which are applied on a per-recording basis:

1. none; no further normalization was applied.
2. normalize per sensor; for each sensor, scale its time-series so that its maximum amplitude is 1.
3. normalize per time step; for each time step, linearly scale the values of all sensors such that the sensor with the largest amplitude has value 1.
4. z-score per sensor; for each sensor, transform its time-series to its z-score, giving it mean $\mu = 0$ and standard deviation $\sigma = 1$.
5. z-score per time step; for each time step, transform all sensor values with the z-score.

7.3.3 *Neural network optimization*

In order to determine the optimal network hyperparameters and preprocessing methods, we perform a parameter sweep for both. We first determine a baseline parameter set which gives a reasonable performance, through trial and error. Then, optimal values for all parameters were found by modifying one variable at a time while keeping others constant, which identifies the effect of that single variable on the performance of the complete pipeline. No grid-wise search is performed due to the excessive time it would take to test each combination.

We further limit the preprocessing parameter sweep to exclude effective time windows that exceed one second, for which we have several reasons. First, longer time windows effectively reduce the amount of training examples, making it harder for generalization to occur. Secondly, longer window sizes may include past data that contains less information and might also be irrelevant for the current location, making it harder to learn temporal patterns from the data. Finally, for time-sensitive tasks, having a smaller window enables a faster response from the neural networks.

7.3.3.1 *LSTM*

The LSTM network was implemented in Python using Keras [24], a deep learning framework running on top of Tensorflow [1]. Our LSTM implementation used the Adagrad optimizer with a clipnorm of 1 [49], which limits calculated gradients such that they cannot exceed a maximum norm of 1, which helps in combatting overfitting.

Data is fed to the LSTM network in batches of windows. Each complete motion in the data set is split up into windows; the default size of the windows is 30 samples, but this value is varied during the parameter sweep. The default window thus has a size of 30×14, for the x- and y- deflections of each of the 7 functioning sensors. Each window is

Table 7.1: Parameter sweep for the LSTM and signal processing. Indicated here are the default parameters, the variation options for each parameter, and finally the optimal choice for each parameter value.

PARAMETER	DEFAULT	OPTIONS	CHOSEN
nodes	50	10, 20, 50, 100, 200	100
learning rate α	0.05	0.1, 0.05, 1E-2, 1E-3, 1E-4	0.05
decay	1E-6	0, 1E-12, 1E-9, 1E-6, 1E-3	1E-9
dropout	0.20	0, 0.1, 0.2, 0.35, 0.5	0
activation function	ReLU	ReLU, sigmoid, tanh	tanh
window size	30	1, 2, 4, 8, 15, 30	15
stride	1	1, 2, 4, 8, 15, 30	2
down sample rate	8	4, 8, 16	16
normalization	4	1, 2, 3, 4, 5	4 (z-score)

accompanied by a single teacher label corresponding to the ground truth location of the source at the end of that window, creating a many-to-one mapping.

During the parameter sweep, we evaluate the LSTM performance through k-fold cross-validation, with $k = 5$; this results in a train/test/validation split of 80/10/10 %. The order of the 3072 recorded motions is shuffled and each complete motion is assigned to one of the sets.

Based on the parameter sweep (table 7.1), the optimal LSTM network architecture has 100 hidden nodes, a learning rate (α) of 0.05, a learning rate decay of 1×10^{-9}, no dropout, and uses the hyperbolic tangent activation function (act. fun.). The signal processing pipeline for the LSTM is described by a window size (win sz.) of 15 samples, a stride of 2, a down sampling rate (ds. rate) of 16 (giving the a sampling frequency of 15.625 Hz) and uses, z-score per sensor normalization (norm.).

7.3.3.2 *OS-ELM*

For the OS-ELM, we similarly divide complete motions from the data set into separate windows, but we also flatten this window to a 1D vector. For the initial training phase, we create a batch of $1.5 \times N$ windows, where N is the number of hidden units. For the sequential training steps, the batch size was set to 64 windows.

Since there is no validation set needed for the OS-ELM training phase, we apply an 80/20 % train/test split. Again, with 5-fold cross validation, we shuffled the order of the data set and assigned each fold to their respective train or test set.

Table 7.2: Parameter sweep for the OS-ELM and and signal processing. Indicated here are the default parameters, the variation options for each parameter, and finally the optimal choice for each parameter value.

PARAMETER	DEFAULT	OPTIONS	CHOSEN
nodes	5000	{1, 2, 5, 10, 20, 30} × 1000	30 000
activation function	sigmoid	ReLU, sigmoid, tanh	ReLU
window size	30	1, 2, 4, 8, 15, 30	15
stride	1	1, 2, 4, 8, 15, 30	1
down sample rate	8	4, 8, 16	16
normalization	4	1, 2, 3, 4, 5	4 (z-score)

Table 7.2 shows the results for the OS-ELM parameter sweep. Based on the parameter sweep, our final ELM network architecture had 30 000 hidden nodes, and uses the ReLU activation function. The processing pipeline includes a window size of 15 samples, a stride of 1, a down sampling factor of 16 (giving a sampling frequency of 15.625 Hz), and uses z-score normalization.

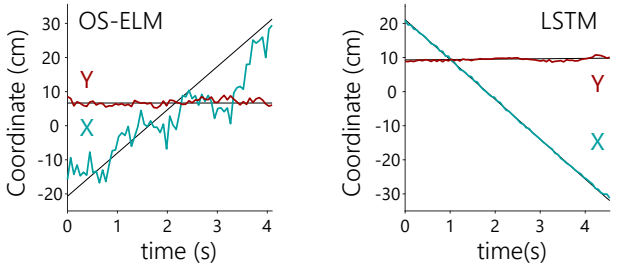
The constant speed section of each motion lasts between 4.3 s for horizontal and 4.5 s for diagonal motion. This results in a data set of $4.4 \times 15.625 \times 3072 \simeq 2.1 \times 10^5$ input examples for the OS-ELM. For the LSTM framework, given the stride of 2 rather than 1, the data set is roughly half that size.

Although the number of hidden nodes for the OS-ELM may seem high, the number of input examples (windows) is roughly 7 times higher. This makes it unlikely for overfitting to occur.

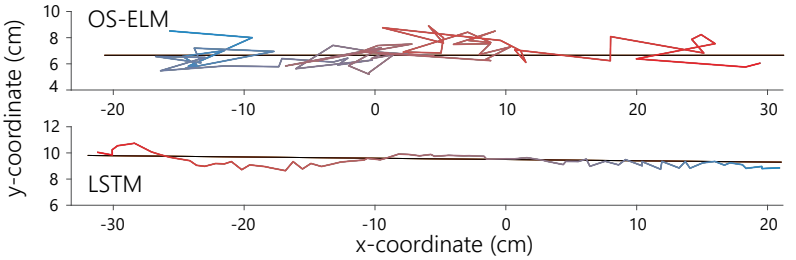
7.3.4 *Influence of noise*

We create a separate data set in order to systematically determine the influence of noise. In particular, we are interested in the localization robustness of the two types of neural networks with respect to different noise levels. This data set consists of 60 repetitions of both backwards and forward motion along two unique paths parallel to the array, at 6.15 cm and 8.15 cm distance respectively, producing 240 recorded motions. These distances are indicated via red dashes in figure 7.3b.

For augmenting this data set, we employ a controlled method of adding noise to our data; we want to stay as close to the real environment as possible, while still being able to quantify how much noise is added. We therefore first record a large data set without any stimulus or plotter activity, from which we sample noise specific to each sensor. We augment the data set by combining recordings with the following nine noise augmentation amplification levels: {0, 0.1, 0.25, 0.5, 1, 1.5, 2, 3, 4}, producing a final



(a) Reconstructed coordinates over time



(b) Reconstructed motion over time

Figure 7.4: Predicted and true labels for one example run. Subfigure (a) shows the true coordinates over time (black lines) as well as the prediction of the OS-ELM and LSTM for the x-coordinate (teal) and y-coordinate (red). Subfigure (b) shows the true path of motion (black lines) and the reconstructed path (time progresses teal to red) for both networks.

augmented set of 2160 motions. This range is selected on a pseudo-logarithmic scale to allow testing a large range of noise levels.

We keep the previously determined optimal hyper parameters for both type of neural networks and thus do not re-optimize for this augmented data set. For determining the performance of each network, we employ 5-fold proportional stratified sampling; each of the five folds has an equal amount of randomly picked complete motions from each of the nine augmented sets. The training data is shuffled before using these folds for training the networks.

7.4 RESULTS

In the first series of experiments, we compared the performance of a feed-forward neural network (OS-ELM) to the performance of a recurrent neural network (LSTM) on the task of localizing a source using measured artificial lateral line (ALL) data.

7.4.1 *Localization performance*

Both networks were optimized with respect to their hyper parameters and preprocessing methods using 5-fold cross validation, as described in the Methods section.

Figure 7.4 shows an example of a single run, where both networks predict the location of an object moving parallel to the array. This figure shows that the LSTM network output is less erratic and closer to the target path compared to the ELM prediction.

When we combine and interpolate between all locations in the region of interest, we can show a spatial mapping of the localization error. Figure 7.5 shows the interpolated error magnitude at each location. From these spatial plots, we observe that the localization error further away from the array increases for both neural networks.

The actual distribution of localization errors, regardless of relative location, is shown in figure 7.6a. Here, we binned the localization error for both types of networks, with a limit of 15 cm. The average and standard deviation of the localization errors are 0.72 ± 1.04 cm and 4.27 ± 3.33 cm for the LSTM and OS-ELM respectively. While the LSTM errors are concentrated around the average, the OS-ELM has a large tail in the distribution (figure 7.6a), with 4.2% of the errors exceeding 15 cm (not shown).

These results show that the LSTM network significantly outperforms the OS-ELM network on this 2D source localization task using measured velocity profiles from a 2D-sensitive array.

7.4.2 *Noise robustness*

To further determine the suitability of both neural networks for other circumstances or other types of artificial lateral lines, we also investigated the effect of noise with respect to localization.

In our performance analysis, we differentiate between noise augmentation levels within the test set. Figure 7.6b and table 7.3 both provide an insight in the effect of added noise to the overall performance. As the noise level increases, so do the LSTM and OS-ELM network's average localization errors, although the LSTM seems relatively less affected by higher noise levels.

Figure 7.6b shows box plots of OS-ELM and LSTM performance, separated per noise level. For all noise levels considered, the median LSTM error is significantly lower than the OS-ELM error.

Table 7.3 shows the overall and per-noise-level error for both networks. On the lowest noise level (no noise), the LSTM achieved an average error of 0.55 cm, whereas the ELM had an average error of 1.75 cm. For the highest noise level ($4\times$), these errors increase to 1.10 cm and 5.64 cm for the LSTM and ELM networks respectively. Again, the error distribution of the OS-ELM shows a long tail towards higher errors, making it less reliable than the LSTM in this comparison.

Table 7.3: Error per noise level for LSTM and OS-ELM networks after stratified 5-fold cross-validation. Shown here are the mean Euclidean error and standard deviation. Finally, the average error over all noise levels is shown.

NOISE LEVEL	LSTM ERROR ($\mu \pm \sigma$ cm)	OS-ELM ERROR ($\mu \pm \sigma$ cm)
0.00	0.55 ± 1.42	1.75 ± 1.52
0.10	0.57 ± 1.60	1.74 ± 1.44
0.25	0.58 ± 1.66	1.92 ± 1.63
0.50	0.57 ±1.51	2.34 ± 1.90
1.00	0.61 ± 1.48	3.00 ± 2.40
1.50	0.66 ± 1.75	3.46 ± 2.65
2.00	0.71 ± 1.75	3.89 ± 3.04
3.00	0.82 ± 1.83	4.70 ± 3.68
4.00	1.10 ± 2.45	5.64 ± 4.37
average	0.69 ± 1.72	3.16 ± 2.51

7.5 DISCUSSION

The aim of our research was to determine how well recurrent neural networks performed compared to feed-forward neural networks, which are the current state of the art, on the task of processing measured artificial lateral line data.

Based on the results from our first series of experiments, we can safely conclude that the recurrent neural network (LSTM) is able to accurately estimate moving underwater source locations. Furthermore, the LSTM network outperformed the OS-ELM network, with an average localization error of 0.72 cm for the LSTM compared to average error of 4.27 cm for the OS-ELM. While this result may not generalize to all feed-forward and recurrent networks, it does indicate that recurrent networks may be more suited to processing ALL data than some feed-forward networks.

Overall, the spatial mapping of localization errors indicate that objects moving at distances further away from the array are harder to localize. This is in agreement with other research and with theory; i.e. the signal to noise ratio of measured and expected velocity profiles negatively correlate with distance.

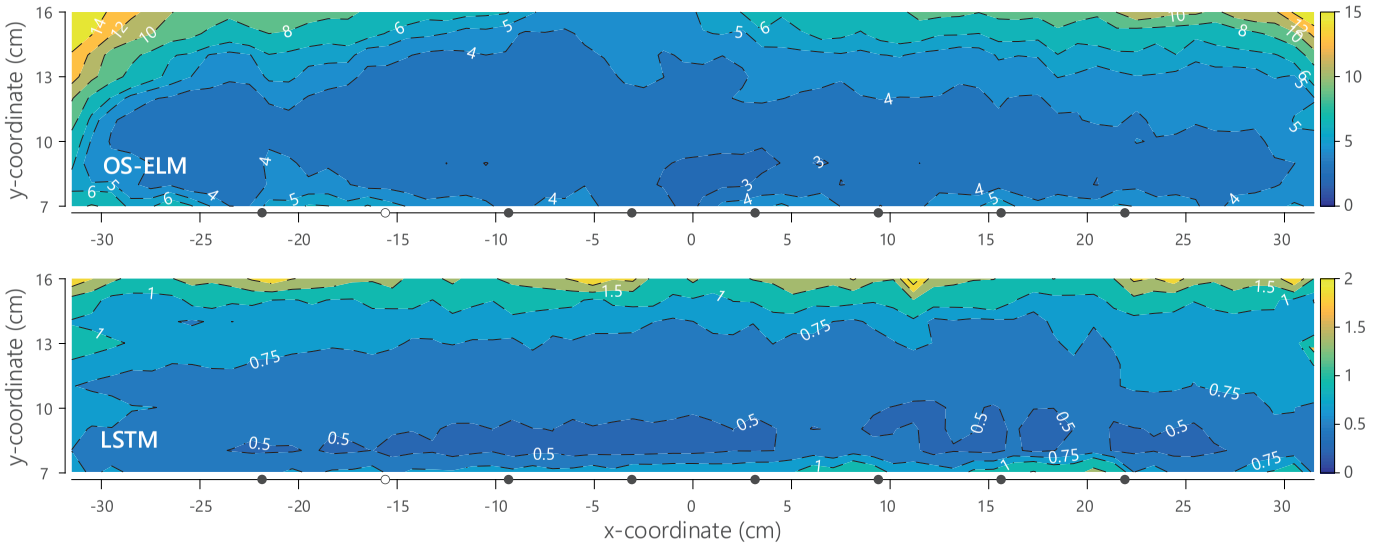
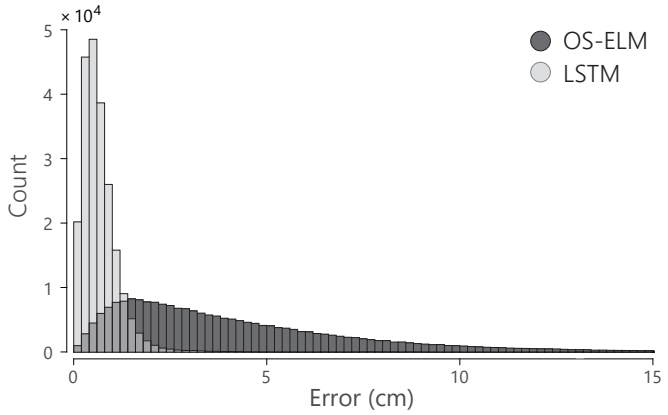
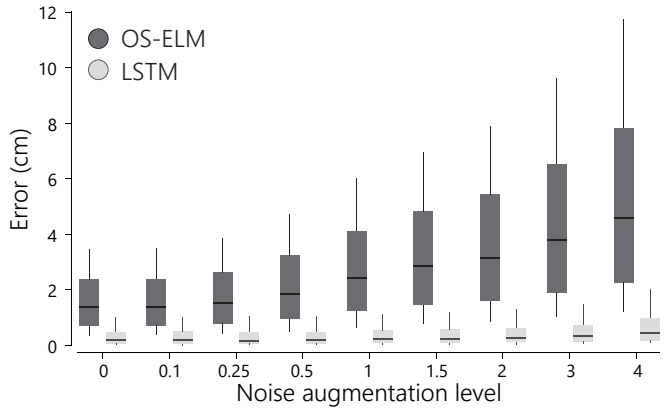


Figure 7.5: Spatial distribution of localization errors (cm) for both types of network. Note that the color bars are scaled per network, up to 15 cm error for the OS-ELM and only up to 2 cm error for the better performing LSTM. The small dots on the x-axis indicate the x-coordinates of each sensor.



(a) Error distribution (first data set)



(b) Error per noise level (second data set)

Figure 7.6: Subfigure (a) shows the error distribution for the first set of experiments, which has a significant different shape for both networks. Subfigure (b) shows the median (bold line) and confidence intervals for different noise augmentation levels. Here, we show 10, 25, 50, 75, and 90 % confidence levels.

The omitted malfunctioning sensor doesn't significantly affect the (local) spatial localization errors as indicated in figure 7.5. On average, the left half of the region of interest (with the omitted sensor) has slightly higher errors. It is unclear why this effect is limited, although it might be a result of the unbiased optimization nature of neural networks; all locations are treated with equal importance. In the case of object detection or obstacle avoidance, nearby locations may be considered more relevant.

A more advanced localization algorithm may consider a weighted optimization error metric, depending on the distance of a target location to each (functioning) sensor in the array, as an approximation to Fisher Information or the Cramer-Rao bound [5].

The signal-to-noise ratio of the measured velocity profile of an object highly depends on its size, speed, and distance with respect to the array. The specific sensor array sensitivity also influences this ratio. The second, augmented data set with varying noise levels allowed us to assess the performance of both network types under influence of sensor noise, independent of object distance. By changing and varying the input noise, we can effectively emulate different (lower) sensor array sensitivities, which makes the findings more applicable to situations with a lower signal to noise ratio. The average localization errors are 0.69 cm and 3.16 cm for the LSTM and OS-ELM networks respectively. Furthermore, while both networks decrease in performance with higher noise levels, the LSTM is clearly less affected, suggesting that recurrent connections may help in alleviating the effects of random, uncorrelated input noise.

Our current sensor array of 45 cm with 7 functional 2D-sensitive all-optical sensors in combination with our LSTM implementation leads to an average localization error of 0.72 cm in a region of interest of 62×9.5 cm. This result cannot be easily directly compared to the state of the art, because other works use other type of sensors and other relative dimensions for their sensor array and region of interest [67, 81]. Furthermore, most studies only consider stationary vibrating (c.f. [3]) or moving vibrating objects [6]; the stimulation frequency can be used to filter the signal and thus produces higher signal-to-noise data than possible for non-vibrating moving objects.

In chapter 6 we showed an 3.3 cm localization error for a moving sphere with a 4-sensor array of 12 cm in an area of 36×11 cm using an ELM [121]. The maximal distance of the current study is similar, while both the region of interest and sensor array are considerably wider. We therefore conclude that the OS-ELM performance is on par while the LSTM implementation outperforms the state of the art for unidirectionally moving objects with constant velocity.

There are two main considerations for future work and expansions to our experiments. First, although our stimulus is more complex than the standard dipole bench mark, the motion was still restricted at one depth-level and in a straight line with constant speed. This type of motion naturally favors taking longer time windows and using memory (i.e. recurrent connections), since the path of the object is predictable.

For future demonstrations of versatile artificial lateral line hydrodynamic imaging, one could consider more complex object propulsion and object shapes. These bench mark stimuli are limited in the sense that they do not emulate fish motion [132] or aquatic propulsion, but rather they provide a controlled, fair comparison method. By using a moving, rather than vibrating, source, we increased the complexity of the hydrodynamic environment by allowing wakes and vortices to be measured by the sensor array. This is however no substitution for detecting an object in freestream flow or in a hydrodynamic

environment with multiple objects or alternative flow sources. Future research could explore localizing self-propelling objects using a stationary artificial lateral line.

Secondly, while the LSTM is quite an advanced network, the OS-ELM is limited in its complexity. Future research could compare more complex feed-forward neural networks or simpler recurrent neural networks, to further investigate the effect of recurrence versus network complexity. In one simulation study [20], the ELM and Echo State Network (ESN) are compared where they have similar levels of performance. In this regard, the ESN is an interesting comparison candidate: it has recurrent connections although the network structure closely resembles the ELM.

7.6 CONCLUSION

In conclusion, a system with recurrent neural networks is capable of hydrodynamic imaging; i.e. localizing a moving underwater source in a 2D plane based on measurements with a 2D-sensitive artificial lateral line of all-optical sensors. This type of passive object sensing does not rely on vision or an active beacon and can therefore work in dark or murky environments. It may therefore, despite its limited range, be favored over traditional detection technologies when considering ecology or covert sensing.

The recurrent network (LSTM) significantly outperforms the feed-forward network (OS-ELM) for localizing a moving object in quiescent water. This may be due to the temporal correlation between subsequent object locations and the resulting subsequent measured velocity patterns. In addition, the recurrent network is relatively less affected by sensing noise, and performs more consistently throughout the area of interest. This indicates that recurrent neural networks may be especially suited for hydrodynamic imaging.

Part IV

SHAPE RECOGNITION

The final demonstration for hydrodynamic imaging involves detecting the shape of a nearby moving object. The following chapter describes a large-scale experiment performed in a swimming pool, featuring measurements from several differently shaped objects.

8

LARGE-SCALE EXPERIMENTS FOR MOVING OBJECT CLASSIFICATION

This chapter of the thesis describes the final demonstration of hydrodynamic imaging. We increase the scale of the artificial lateral line itself, as well as the objects used during the experiments. Whereas in part III we demonstrated various methods of object localization, we here determine a different property of the nearby object in this chapter: the object's shape.

Altogether, five objects with different shapes were towed across a swimming pool, with a 3.5 m eight-sensor array mounted at the side of said pool. In this study, we show two methods to process the data coming from the artificial lateral line.

A first processing method is to align the measured sensor data to visualize *hydrodynamic signatures*, which might reflect an objects shape and may therefore provide some insight in the hydrodynamic character of an object.

The second method to process the data is a time window based classification. We first extract physical properties from the measured velocity profiles in a time window, thus constructing location-invariant feature representations for each object. These feature representations are then used to train an ELM classifier to identify an objects shape, regardless of location, distance, and direction of the object.

PROVENANCE A version of this chapter has been published as:
B.J. Wolf, P. Pirih, M. Kruusmaa, S.M. van Netten. Shape classification using hydrodynamic detection via a sparse large-scale 2D-sensitive artificial lateral line. *IEEE Access* 8 11393-11404.

AUTHOR CONTRIBUTIONS BW conceived and supervised the study, built the sensor setup, performed the analysis, wrote the draft, and wrote the final manuscript. BW, PP, and SvN built the moving platform setup. BW, PP, MK, and SvN performed the experiments. PP, MK, and SvN revised the draft.

ABSTRACT Artificial lateral lines are fluid flow sensor arrays, bio-inspired by the fish lateral line organ, that measure a local hydrodynamic environment. These arrays are used to detect objects in water, without relying on light, sound, or on an active beacon. This passive sensing method, called hydrodynamic imaging, is complementary to sonar and vision systems and is suitable for collision avoidance and near-field covert sensing. This sensing method has so far been demonstrated on a biological scale from several to tens of centimeters. Here, we present measurements using a large-scale artificial lateral line of 3.5 meters, consisting of eight all-optical 2D-sensitive flow sensors. We measure the fluid flow as produced by the motion of five different objects, towed across a swimming pool. This results in repeatable stimuli, whose measurements demonstrate a complementary aspect of 2D-sensing. These measurements are both used for constructing temporal hydrodynamic signatures, which reflect the object's shape, and for flow-feature based near-field object classification. For the latter, we present a location-invariant feature extraction method which, using an Extreme Learning Machine neural network, results in a classification F1-score up to 98.6% with selected flow features. We find that, compared to the traditional sensing dimension parallel to the sensor array, the novel transverse fluid velocity component bears more information about the object shape. The classification of objects via hydrodynamic imaging thus benefits from 2D-sensing and can be scaled up to a supra biological scale of several meters.

8.1 INTRODUCTION

The lateral line is a fluid flow sensing organ found in fish, that augments their perception [36]. This organ consists of arrays of fluid flow sensors called neuromasts, located on the skin (superficial neuromasts) or in sub-dermal canals (canal neuromasts), each tailored to sensing different properties of the flow [18]. These neuromasts allow fish to sense and act on their local near-field flow situation. They use this sensation for behaviors such as schooling, and detection of both prey and predators [93]. This biological near-field flow sense is often referred to as *touch at a distance* [36]. It does not rely on ambient light and is therefore more reliable than vision in dark or murky environments.

The lateral line has been used as inspiration to create several biomimetic sensor arrays, called artificial lateral lines (ALLs) [81]. The ALL can be either used as a stationary sensor array or, similar to the biological lateral line, attached to a moving platform, depending on the intended detection task. In both cases, the ALL measures the hydrodynamic effect of relative fluid motion. A stationary ALL can be used for object detection and tracking vibrating dipole sources [4, 12, 31, 125, 131] or moving objects [43, 121, 123], with potential use cases including tracking nearby moving vessels. An ALL attached to a vessel can be used for aligning with the freestream flow [35, 73], or for obstacle avoidance [76, 117], potentially enabling safe navigation for ships or autonomous underwater vehicles (AUVs) [73].

While far-field detection methods such as active sonar can detect ships and other objects, and can resolve their shape over large distances, they rely on an active emitter. Near-field hydrodynamic imaging only works in a short range, but can be more accurate in this range. Furthermore, it is completely passive, enabling covert sensing, and is unaffected by murky waters, an advantage over camera based and sound based solutions. ALL hydrodynamic imaging thus provides a complementary detection method to vision-based and sonar-based systems [125].

Identifying up to three object shapes with an ALL has been achieved via the wake of an object in controlled flow conditions [25, 70, 82, 106, 111]. There the object is usually placed upstream relative to a sensor array. The resulting wake is repetitive in nature and its periodic features are used for determining the size and shape of objects. An alternative approach [40] assumes still water, where an object moves past the array and creates its own near-field flow. We use this second approach to classify five different objects via their self-produced near-field flow.

The typical length of an ALL ranges from several to tens of centimeters, matching the biological scale of the fish lateral line [81]. The present work describes the first use of a large-scale ALL of several meters and its consequences and adaptations to fluid flow sensor array signal processing. We use eight optical 2D-sensitive flow sensors [124] in a 3.5 m array, to measure the object-produced near-field flow.

We identify key features of the fluid flow for processing the ALL measurements that make object shape classification more robust and less prone to overfitting. There are several automatic feature selection algorithms that can be used for this purpose [54]. Specifically, we employ manual feature selection, filter methods, and wrapper feature selection to determine subsets of features. Then, we use an Extreme Learning Machine (ELM) neural network architecture [64] with cross validation that uses these selected features as input for the final classification step. We also discuss the merits of sparse spatial sampling for constructing hydrodynamic signatures of shapes, which can be used for further analysis and interpretation.

In the next section, we present background information on hydrodynamic imaging and the ELM neural network. In section 3, we describe the experimental setup and the processing pipeline. The fourth section lists the results, which are further discussed and interpreted in section 5. We end with a conclusion section, summarizing our main findings.

8.2 BACKGROUND

8.2.1 *Hydrodynamic imaging*

Hydrodynamic imaging [30, 36, 125] is a bio-inspired method which usually applies to sensor arrays that measure a projection of the local hydrodynamic environment in response to a flow source or moving object, see e.g. figure 8.1. In fish lateral line research,

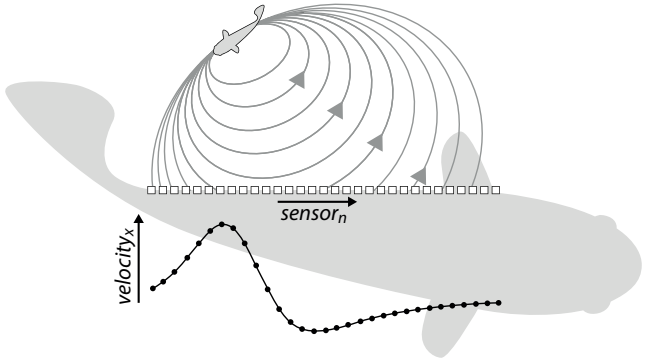


Figure 8.1: Passive hydrodynamic near-field imaging by the fish lateral line. The smaller fish creates a flow field which is sensed by the sensors on the larger fish's body. The instantaneous 1D spatial velocity profile encodes the source's relative location and other properties. Adapted with permission from [122].

the excitation of neuromasts along the trunk, concatenating to a spatial excitation pattern, have been shown to encode the pressure gradient along said trunk [18, 31, 43]. Some fish species have been shown to be capable of detecting objects via hydrodynamic imaging up to a distance of roughly one body length away [26, 31, 69].

An ALL is usually composed of uniformly spaced pressure or fluid velocity sensors [81], sampling the hydrodynamic environment at discrete locations. The concatenation of these discrete samples forms a spatial velocity profile, similar to the spatial excitation patterns measured along the trunk of the fish.

8.2.1.1 *Velocity profiles*

With a simplified hydrodynamic model, called inviscid flow [78, 110] as described in [5, 31, 43, 121], we can estimate the velocity profile as sampled by an array for a moving or vibrating object; this is called the forward problem. This velocity profile encodes state information such as the relative location of an object, its shape, size, speed, and direction [19, 31]. Having such a forward hydrodynamic model does not directly allow for decoding the velocity profile, i.e. determine the mentioned properties of an object. This defines the inverse problem: reconstructing an object's properties from measured velocity profiles.

One aspect of these spatial velocity profiles, notably spatial broadening [31, 121], has been shown to be beneficial with respect to localizing vibrating spheres, a subset of the full inverse problem. The spatial broadening property entails that the generic shape of a velocity profile is consistent when an object is moving in a certain direction. This generic velocity profile shape, or signature, is only scaled and translated by changes in

the relative position, size and speed of the object. This property has been successfully used in template matching methods for source localization [31, 96]. More directly, zero crossings, maxima, and minima of the velocity profile also directly produce estimates for the distance (y-coordinate) and the lateral position (x-coordinate) [43]. In [9, 77, 122], it is shown that an ALL comprising of 2D-sensitive sensors makes localization more robust. This extension compared to the 1D sensitivity of fish and most other ALL implementations, provides more and complementary spatial reference points [9, 122], which may also be helpful for shape recognition.

8.2.1.2 *Hydrodynamic signatures*

Some ALL systems focus on estimating fluid flow parameters such as the flow speed, direction and vorticity, either in natural environments [101] or in confined flow tanks. In the latter case, an object is usually placed upstream with respect to a sensor array [82]. At constant flow rates, this object can shed Kármán vortex streets that encode object characteristics in the spatial and periodic properties of the vortices [19, 25, 70, 111].

The alternative approach, that we use in the current study for object shape classification, is to move the object rather than the medium. This allows measuring the hydrodynamic effects of the object shape.

In the hydrodynamic near-field, the shape of the object affects the pressure gradient and thus also the principal shape of the measured velocity profile. This measured velocity profile shape can be modeled via a process called conformal mapping [19], although these velocity profiles are expected to be less distinctive at distances further from the array [106] and in the limit resemble the signature of a sphere.

So in order to preserve details and identify objects based on their measured velocity profiles, they need to be measured up close. This allows constructing what we define as hydrodynamic signatures of each object in detail.

8.2.1.3 *Flow-based shape classification*

There are several works that have demonstrated object shape classification in simulation or small-scale experiments.

Recently [77], neural networks have been used in a simulation study to determine the shape parameters of a foil-shape object. Using a grid of sensors and a conformal mapping potential flow fluid model, they simulated the x and y fluid flow components, as well as the absolute fluid speed magnitude and the dynamic pressure at each of the sensors. The shape of the object was reconstructed to a high degree via a triplet of neural-network estimated shape parameters. The dynamic pressure magnitude was shown to the best choice of the four considered types of input to the neural network.

In [82], a pressure sensor array is used to determine the shape parameters of a cylinder and cube object in a fluid flow channel. From their reported statistics, we infer an F1-score

of 97.1% for this two-class problem. In a similar study [70], two object shapes (semi-circle and rectangle) of three sizes were analyzed via their wake. Visualizing the dominant wavelength of the vortex streets versus its magnitude reveals that measurements on objects of similar size cluster together. Discerning the shape would however require additional features.

In [40], a stationary array was used in quiescent water to infer the shape parameters of an elliptical cylinder. The objects were moved past the array at a constant speed of 50 cm/s at a constant distance between .5 and 1 cm. Between runs, the angle relative to the array and radii of the elliptical cylinder were varied. By employing a particle filtering method, they were partly able to reconstruct these two types of parameters.

Compared to these related works, we increase the distance and size of the objects, the number of different object shapes, and the scale of the array. In addition, we measure the flow field in two dimensions; novel to the application of shape recognition via hydrodynamic fingerprinting.

8.2.1.4 *Temporal velocity profile sensing*

With small scale or high density sensor arrays, neighboring sensors often show correlation which is harnessed in the signal processing pipeline. This assumes that the sampled velocity profile has an adequate spatial resolution to be matched to a spatial hydrodynamic signature of a known object. For large-scale arrays and applications, this correlation assumption breaks down.

Instead of treating the measurements from the sensor array as under-sampled spatial velocity profiles, we switch to the time-domain and focus on temporal velocity profiles, in our case measured from an object that moves with constant speed past the sensor array. While additional sensors would likely improve the quality of the measurements, one beneficial consequence of sparse spatial sampling is that we have true independent measurements of an event, where hydrodynamic noise can thus be factored out more easily. In fact; we have eight independent measurements across the array.

8.2.2 *ELM neural network*

For classification, we employ a light-weight regression artificial neural network, the Extreme Learning Machine (ELM) [64], which has been shown to be fast and straightforward to optimize for ALL localization [20, 121] without overfitting.

While other, more advanced neural networks might be more powerful, determining optimal network configurations and preventing overfitting for these more complex neural network architectures is not trivial. The ELM has only one network parameter to choose: its hidden layer size, i.e. the amount of hidden neurons, which makes it straightforward to tune to avoid overfitting.

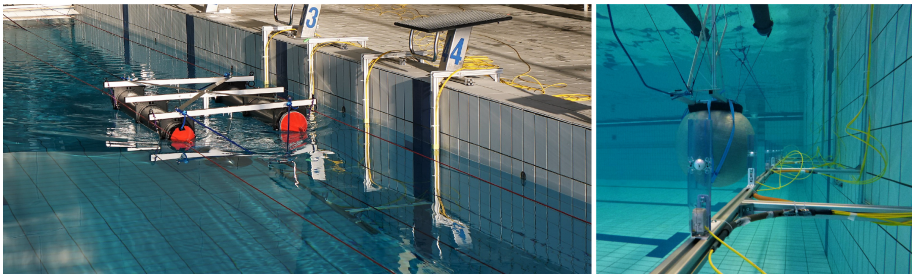


Figure 8.2: Overview of the deployed setup. The left picture shows the towing platform in motion near the center of sensor array with two (red) guiding wires and a partly submerged (blue) towing cable. The right picture shows the sensor array with the white spherical elements at a depth of 140 cm with respect to the water level, as well as the ball object attached to the towing platform.

The second benefit of the ELM neural network is that it can be trained in a fraction of the time needed for an MLP (multi-layer perceptron) or deep neural network. The ELM has only one hidden layer, with weighted connections to the input layer and output layer. The fast training of this network is a result of only the output weights being trained; the input weights remain fixed after initialization [64]. This allows the network to be trained in a single learning step, while still providing a non-linear transformation of the input space.

8.3 METHODS

We measure temporal velocity profiles using a large-scale ALL in response to differently shaped objects attached to a towing platform. These profiles are used to visualize hydrodynamic signatures, as well as input for the classification pipeline. In this pipeline (see Fig. 8.4), we first calculate features, then determine feature subsets, and finally evaluate the classification performance using these subsets via the ELM neural network. These processes are described in detail in the following subsections.

8.3.1 *Setup*

Here, we first describe the sensor array, followed by the towed objects, and finally the experimental procedure.

8.3.1.1 *ALL sensor array*

To demonstrate object shape classification, we installed eight 2D-sensitive all-optical flow sensors [123, 124] each 0.5 m apart at the short side of a 18×25 m swimming pool,

see also figure 8.2. The 3.5 m array was placed 0.5 m from the wall and centered with respect to the 18 m wide pool area. The sensor array has an adjustable submerged depth, which defaults at 1.4 m. The array is adjusted in depth for every submerged object to be level with its center, as to accurately measure the hydrodynamic signature of each object.

Each all-optical flow sensor is constructed using a fiber structure comprising four optical fibers. Since they are all-optical, they do not directly require electricity to function, but rather operate on optical signals using fiber Bragg gratings (FBGs) [124]. The white spherical element at the tip of each sensor (figure 8.2, right) picks up fluid forces and deflects the fiber structure. This deflection is measured using the FBGs inside the fiber structure which can be read out at a distance via optical cables.

The 32 optical signals from 8 sensors were simultaneously measured using a Hyperion si-225 optical interrogator (Micron Optics, USA) at 5 kHz, and downsampled to 200Hz. We report and graph the sensed local fluid velocity for each of these sensors in the x-dimension (parallel to the array) and y-dimension (orthogonal to the array).

Each sensor is encased by a perforated 32 mm diameter tube to protect the sensors from possible impact. These protecting tubes slightly lower the resonance frequency and sensitivity of the sensors, but ensure safe and prolonged operation [123].

8.3.1.2 *Towed objects*

We measure the temporal velocity profiles resulting from different objects towed past the ALL. In addition, for a null measurement baseline, we measured velocity profiles with only the towing platform as the *no object* case.

The towing platform is constructed from two floating pontoons 70 cm's apart, each 16 cm in diameter and 150 cm in length, together providing 60 L of buoyant volume. An electric winch was used to tow the platform and attached object with a constant speed, specified to maximally tow 200 kg at 0.3 m/s.

Five differently shaped objects were chosen for the experiments (figure 8.3). A dumbbell and barrel shape were chosen since they displace roughly equal amounts of water, yet have a different shape. A ball was chosen as it matches the spherical shape used in theory and most ALL characterization efforts [81]; moreover it displaces a lot of water, producing a strong signal. Finally, two variants of a capsule shape were deployed. First a slender capsule, and secondly the same capsule with an attached inverted bowl, which acts as a wake-shedder. All shapes were mounted a meter under the towing platform and were made neutrally buoyant by filling them with water. Additional schematic side views of the objects and the towing platform can be found in the supplementary material, figure 8.S1.

Table 8.1: List of run distances, object sizes, and estimated Reynolds numbers (Re) based on their displaced volume.

OBJECT	DISTANCES (m)	SIZE (cm)	REYNOLDS NO.
Baseline	0.4	-	-
Barrel	0.4, 0.57	⊘ 37 × 63	15×10^4
Ball	0.6, 0.84	⊘ 70	21×10^4
Dumbbell	0.4, 0.57	⊘ 35 × 70	14×10^4
Capsule	0.4	⊘ 28 × 185	18×10^4
C. + shedder	0.4	⊘ 28 × (185 + 30)	19×10^4

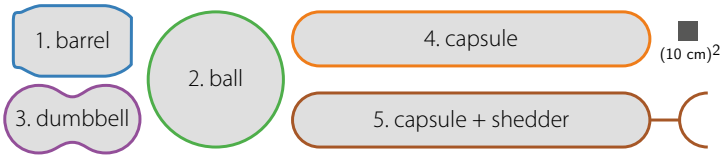


Figure 8.3: Schematic side view (to scale) of the objects used.

8.3.1.3 *Experimental procedure*

Throughout the rest of this chapter, we consider a *run* to be a completed motion of the towing platform, towing an object past the array once. For each object, we measure three runs from left to right (forwards) and three runs from right to left (backwards) at each distance as indicated in table 8.1. With an average run duration of 50 seconds to cross the pool, we selected 20 seconds from each run in which the towing platform passes by the array.

The inviscid hydrodynamic model [78, 110] as described for 2D sensing in [121] allows determining an upper bound on the detection distance, depending on the stimulus in question. For a conservative estimate, we consider a 0.2 m radius object moving parallel to the array at 0.3 m/s, and a noise-level equivalent sensor threshold of 5 mm/s for DC signals [124]. For a signal to noise ratio of one, such an object can be detected with the center at 0.62 m distance from the sensor array. To preserve more details for the hydrodynamic signature of these objects, we vary the object center distances around a closer value, see also table 8.1.

Table 8.2: List of used features which are calculated separately for the x and y dimensions, combining to 48 features in total.

FEATURE	DESCRIPTION	AMOUNT
DC	0 Hz	1
midband frequency of f-bands	{.25, .35, .50, .70, 1.0, 1.4, 2.0, 2.8, 4.0, 5.7, 8.0, 11, 16, 23, 32, 45} Hz	16
centroid f	-	1
kurtosis	{min ($\vee$), max ($\wedge$), σ}	3
skewness	{min ($\vee$), max ($\wedge$), σ}	3

8.3.2 *Signal processing*

We process the data in two ways. First we align and combine all temporal velocity profiles to visualize temporal hydrodynamic signatures. In contrast, for the object classification framework, we use the eight unaligned and unfiltered 2D velocity profiles.

8.3.2.1 *Aligning for hydrodynamic signatures*

To construct the signatures, we combine 24 temporal velocity profiles from 8 sensors and 3 repeated runs for each object, distance, direction, and sensing dimension. We first low-pass filter the signals (Butterworth, 3 Hz) and use correlation to find the delays between sensors. Then, we time-shift the original velocity profiles to correct for these delays. Finally, we show the median signature, as well as the 25 to 75 percentiles as an indication of the consistency of the measured signatures. For an example of this process, see the left pane in figure 8.4.

8.3.2.2 *Feature extraction*

In our classification pipeline, we make use of a moving window approach. Here we use a time window of 4 seconds and a stride of 1 second to calculate features. This length was chosen since characteristic wakes and hydrodynamic stimuli are in the <5 Hz range and are well-captured in this time-scale. For all features, we calculate the feature values for the x and y sensing dimension separately.

We consider two types of features, spectrum based and distribution based. We briefly describe them here, an overview can be found in table 8.2.

We include the DC offset (mm/s) within a time window as the first spectral feature. The energy in different frequency bands proved to be an important feature for classifying

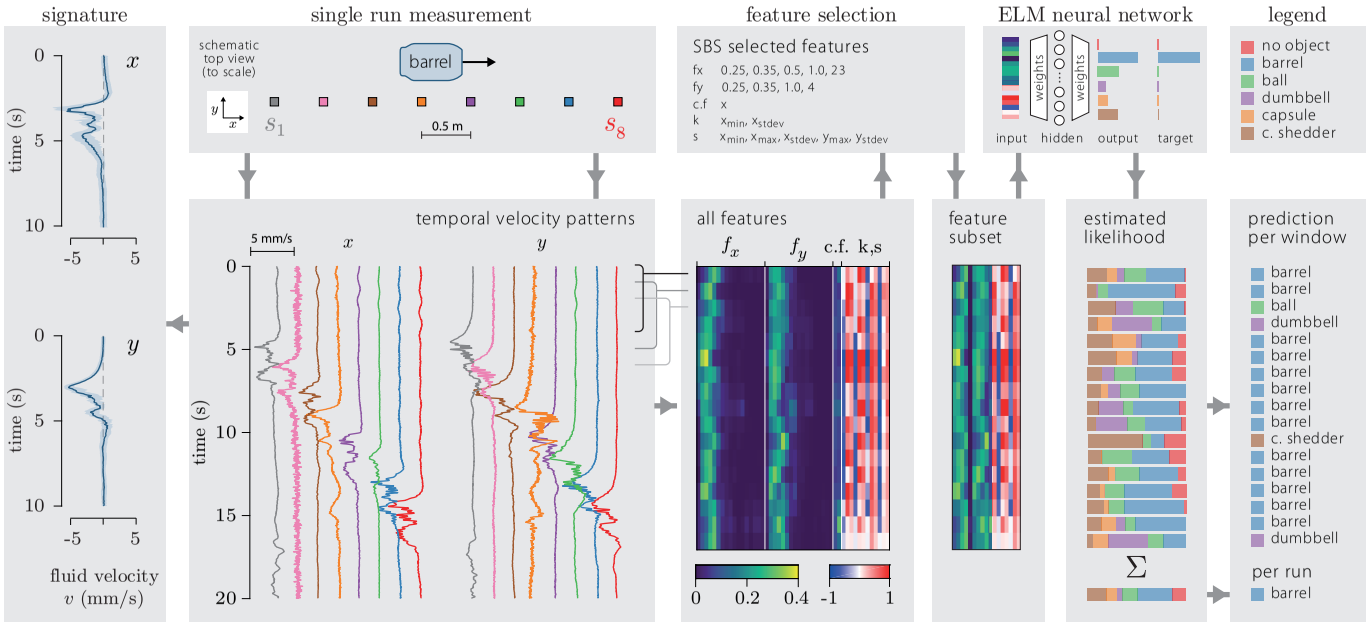


Figure 8.4: Signature and classification pipeline overview for a forward moving barrel. Arrows between the blocks show the flow of signal processing. The single run is measured by the ALL and yields temporal velocity profiles for both sensing dimensions. These are time-shifted and aligned to produce a signature in the leftmost pane. For classification, we calculate features based on overlapping 4 second windows. Here, $f_{x,y}$ denote the frequency bands, `c.f.` the centroid frequencies, and `k` and `s` the kurtosis and skewness features respectively. We use feature selection methods to create feature subsets, the result of the SBS method is shown here. The ELM is trained to match each feature (input) vector to a target vector indicating the correct class. When the trained ELM is presented with a new feature vector as input, the network outputs likelihoods for each class. By selecting the maximal likelihood for each time window, or for the whole run, we obtain shape predictions.

river flow conditions [101]. We therefore select 16 frequency bands from the ANSI half-octave band definition [65], using 0.25 Hz as a reference frequency. This yields mid band frequencies from 0.25 Hz to 45 Hz on a logarithmic scale. We used 16 2nd-order Butterworth filters to filter each windowed signal and determine the frequency band energies. The final spectral feature we use is the centroid frequency of the average spectrum in a time window. For all spectral features, we average the feature value over the sensors to a single feature value per time window.

The second feature type, i.e. kurtosis and skewness, describes the distribution of data in the time windows, as used before in fluid flow classification [101]. Instead of averaging over the eight sensors, we take the minimal value, maximal value and standard deviation of both as separate features.

8.3.2.3 *Feature normalization*

Before selecting the features and feeding them to the classifier (section 8.3.3.1), the feature vectors are normalized to have their feature values in a more favorable range for the classifier.

For the frequency bands, we take the spectral power for each band and divide this by the bandwidth of their respective band. We further normalize the frequency band features for each time window by dividing their values by the total power for that time window. This process re-scales all values to the range from 0 to 1 and makes their sum 1. These features are now less dependent on the actual flow speed, but rather encode relative contributions, making them more versatile with respect to object size.

For kurtosis and skewness we selected the minima, maxima and standard deviation over the eight sensors. We normalize these by applying the hyperbolic tangent, scaling these distribution feature values between -1 and 1.

8.3.2.4 *Feature selection*

Some of the selected features might still be irrelevant or encode duplicate information. To aid the classifier in preventing overfitting, we search for subsets of features that are relevant for discerning the objects. We consider three types of methods for selecting feature subsets [54]: manual selection, filter methods, and wrapper methods. By showing the performance of these subsets, we can infer which features are more informative for classifying objects.

We first demonstrate the performance of the classifier with handpicked subsets, in our case: all 48 features, frequency bands, only x frequency bands, only y frequency bands, and finally the centroid frequencies combined with kurtosis and skewness.

Filter methods usually select features based on their similarity or relevance, defined through correlation. These methods are unsupervised, in the sense that they do not make use of the information which class certain feature vectors belong to. For the filter

methods, we use Lasso, reliefF [102], FCBF [129], mRMR [55], and cmRMR [37]. Two of these methods require the features to be discretized in bins, others work on continuous values. When binning is required, we define 16 sequential bins per feature, all equally populated. This is achieved via determining its distribution, and splitting the distribution on the median of the feature. We repeat this step four times to obtain the bin boundaries. We only use binning for selecting the features; for the final classification, the original continuous feature values are used.

Wrapper methods usually test different combinations of features and monitor the classification performance; a supervised method. For the wrapper method, we use a $\ell 1$-svm (support vector machine) as a linear classifier, as suggested by Tang, Alelyani, and Liu [114] to provide an unbiased wrapper. Were one to use the final classifier, one could tailor the feature selection process and thus cause overfitting, reducing generalization. With this wrapper, we perform both a sequential backward search (SBS), eliminating one feature at a time; and a sequential forward search (SFS), adding one feature at a time, to produce a subset of features.

8.3.3 *Classification*

We measure three repeated runs per object, distance, and motion direction, resulting in a total of 54 measured runs. We employ 3-fold stratified cross validation: each fold contains a complete set of 18 runs. Two folds are used for training and tuning the neural network, whereas the leftover fold is used for testing.

8.3.3.1 *ELM neural network training*

As described in the Background section, the ELM has a single tunable parameter: the size of the hidden layer. To determine the optimal size before each training phase, we employ a nested validation scheme. During this tuning-phase, the data in the training set is further randomly divided into 5 sub-folds for nested cross validation, four of which are used for training the network. We use the remaining sub-fold in this tuning phase for monitoring the performance while varying the network hidden layer size.

For the training phase, we then take the five found optimal sizes during the tuning phase and use the average of these sizes to retrain the network on the whole training set. When the network is trained, we test it on the remaining fold for the exploitation phase.

As for the network structure, the trained network's input size is determined by the number of selected features. Its hidden layer size is determined during the tuning phase, which was usually between 40 and 50 nodes. The output is always a 6-element vector: a likelihood score for each class, also known as one-hot encoding.

8.3.3.2 *Classifier performance*

Because there is a separate repetition of each type of run in each fold, we can assert the performance consistency. Defined as usual, we report the classification performance as the precision, recall, and average F1-score. We have two modes of scoring the classifier performance.

First, we report the classification performance *per window*. Here we simply take the maximal value of the output vector as the predicted object for each time window.

The second mode of reporting, *per run*, first averages the output vectors of the 17 time windows in a run, then taking the maximal value as a single prediction for the whole run. With this approach, our classification pipeline has the attractive property that it can use mounting evidence and can thus be used on signals with arbitrary length.

8.4 RESULTS

We first discuss the constructed temporal hydrodynamic signatures and reflect on their apparent relation to the object shapes. Secondly, we report the shape classification performance for different feature subsets, which indicate what types of features are suited for discriminating between object shapes.

8.4.1 *Temporal hydrodynamic signatures*

We measured the velocity profiles resulting from moving five different objects past our array. Including the empty towing platform as a baseline, we have a total of six classes. Figure 8.5 shows constructed signatures, as well as the theoretical signature for a sphere in inviscid fluid [110] as described for 2D in chapter 5. We show the x- and y-component separately and make a distinction between the direction of motion and the measured distances.

We measured some objects at two distances, to show the effect of distance on the measured hydrodynamic signatures. As described in section 8.2.1.1, the signatures at the further distances are indeed lower in magnitude and lose detail compared to their respective nearby version. When further away, these object signatures start to resemble the signature of a sphere, losing higher order components to the noise. The objects are still detectable, but not immediately identifiable with respect to their shape.

The complementary effect of 2D sensing is also visible in these temporal hydrodynamic signatures. When e.g. the x sensing value reaches a zero-crossing, the y dimension often shows a local extrema or other non-zero value and vice versa. This increases the signal to noise ratio of the combined measurement at these informative points.

Overall, the signatures resulting from forward and backward motion are quite similar in shape. The notable exceptions here are the dumbbell object and the capsule shedder object. The asymmetry in the dumbbell's signatures is likely caused by the object being

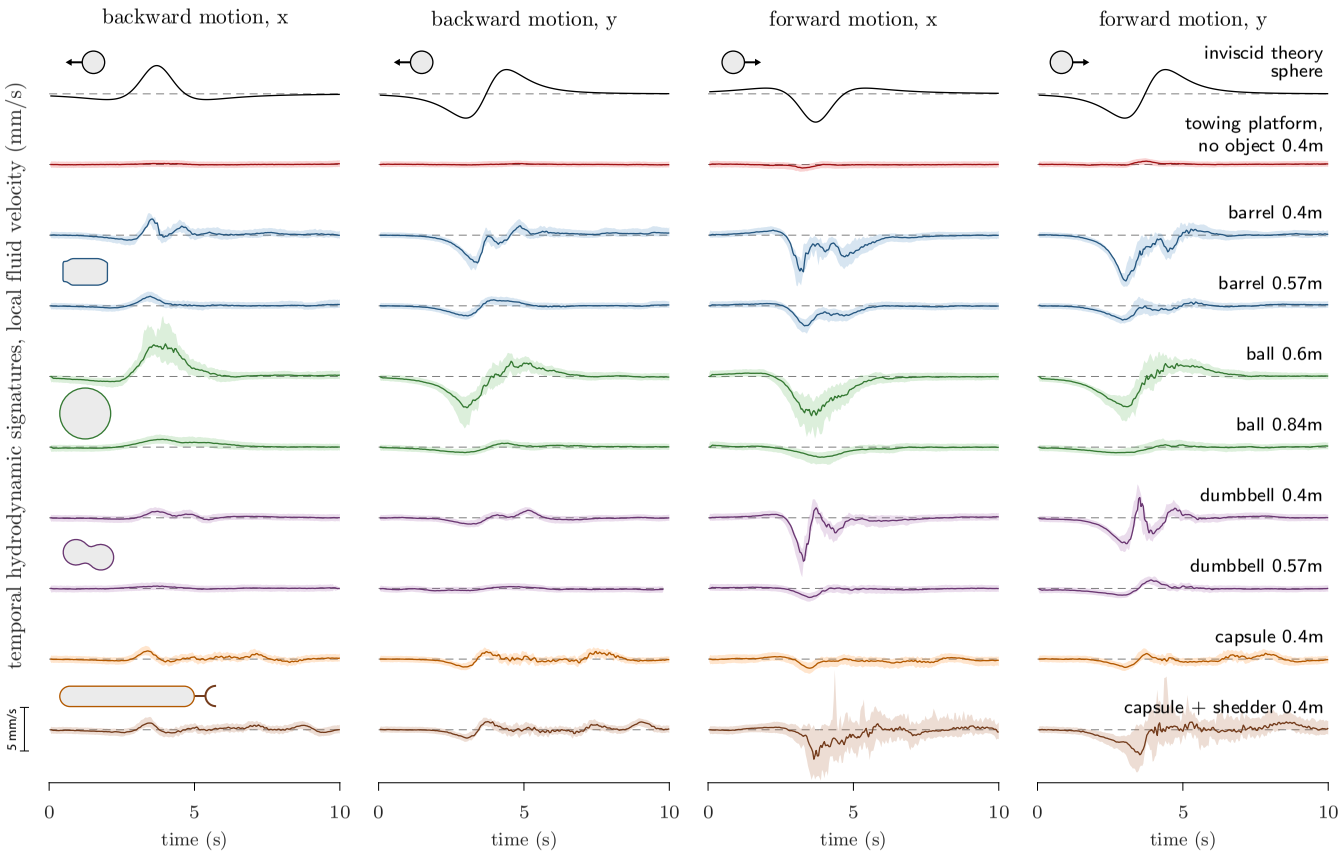


Figure 8.5: Constructed temporal hydrodynamic signatures of each object and distance (right pane). The object shapes are indicated to relative scale in the left pane. The signatures are visualized with the median response and the 25th to 75th percentiles (fills), which are calculated from 24 time-aligned temporal velocity profiles.

mounted at a slight angle. For the capsule shedder object, the forward motion clearly causes turbulent wake shedding to occur.

Furthermore, several identifiable relations emerge between the object shapes and their measured hydrodynamic signatures (figure 8.5). First, the ball-shaped object produces a signature similar to that of a modeled sphere in inviscid flow. A second relation is visible between the barrel and dumbbell shape. Both signatures are similar in that they show two minima and maxima, where the dumbbell has a sharper signature, especially during forward motion. Thirdly, the effect of the wake shedder on the capsule is clearly visible during the forward motion, producing turbulent behavior. In addition, during backward motion, the signatures of these two shapes are similar and match quite well. They differ in a subtle extra bump at the 9 second mark, coinciding with the addition of the inverted bowl.

These observations suggest that the object shape is reflected in their temporal hydrodynamic signature.

8.4.2 *Shape classification*

For shape identification, we only considered the original unaligned sensor measurements and the object shape, ignoring the distance and relative location of the object.

8.4.2.1 *Classification pipeline*

One example of the process of a classified run is visualized in figure 8.4. From the visualized feature vector, we observe that the energy bands in the lower part of the spectrum contain the most energy, as expected. Furthermore, the spectral feature values remain similar throughout the subsequent time windows of the run. A detailed version of the feature space can be found in the supplementary information, figure 8.S2.

For for 4 out of 17 time windows in this example run (figure 8.4), the barrel object did not receive the highest neural network output. If we however aggregate this output over the whole run, the barrel shape clearly has the highest predicted likelihood and is correctly predicted as a barrel using the *per run* approach.

8.4.2.2 *Feature importance*

Figure 8.6 provides a visual indication of the classification performance for each subset, as well as the exact achieved classification test score. A full list of chosen features for each feature subset (table 8.S1) and exact scores (table 8.S2) can be found in the supplementary material.

The influence of each feature subsets on overfitting is also visible in figure 8.6 via the discrepancy between *train* and *test* performance. A large discrepancy indicates

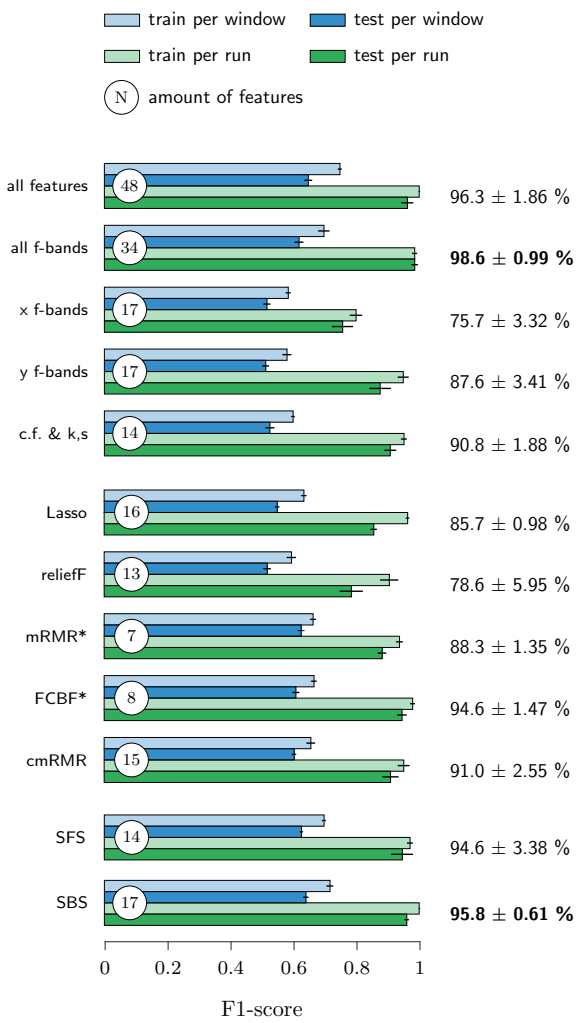


Figure 8.6: Bar plots indicating the 3-fold classification performance on the train and test sets for several feature subsets. Handpicked sets are grouped left, filter method found sets are grouped right, and the two wrappers are displayed at the bottom. The test *per run* score and standard deviation are displayed next to each bar plot, other scores are included as supplementary information (table 8.S2). The best performing handpicked and algorithmically determined subset are indicated in bold. Methods with a $*$ make use of a discretized version of the features during the selection.

overfitting and loss of generalization. Quite notable is the second feature subset 'all f-bands', where seemingly no overfitting takes place.

The performance with the handpicked feature subsets, as indicated figure 8.6, show that the spectrum features (all f-bands) yield higher performance than the other features combined (c.f. & k,s). In addition, the velocity profile orthogonal to the array, y, seems more informative for discerning the objects, compared to the traditional parallel component x, as sensed by the fish lateral line and as used in most 1D ALL applications [81].

For the algorithmically generated subsets, the quantitative contribution of each feature to the final classification performance is hard to estimate, given that one also has to consider the interaction and overlap between the features. Some feature selection algorithms provided subsets of features which led to poor performance consistency, such as the Lasso and reliefF methods. These methods provided subsets that perform well on the training set, but show a considerable drop in performance for the test set, which indicates poor generalization.

For our experiments, the handpicked 'all f-bands' subset (98.6 %) and the ℓ_1-SBS (sequential backward search, 95.8 %) algorithm provide the best feature subsets that show both a high testing performance as well as a low discrepancy between training and testing performance, indicating good generalization.

8.5 DISCUSSION

The sensor array of optical 2D-sensitive flow sensors allowed us to capture a profile of the hydrodynamic environment at eight measurement points in 2D. We have demonstrated that the sparse large-scale 2D-sensitive ALL enables hydrodynamic shape identification with high precision.

By carefully optimizing the ELM neural network classifier via nested cross-validation and employing feature selection, we kept overfitting to a minimum, as signified by the small discrepancies between train and test scores. This indicates that the methods and features used may generalize well to different sets of measurements.

8.5.1 *Hydrodynamic signatures*

As mentioned in the introduction and background sections, our sparse large-scale ALL has the benefit of independently sampling eight temporal velocity profiles. We realigned these profiles to construct temporal hydrodynamic signatures (figure 8.5), which can be used for further analysis.

The temporal hydrodynamic signatures show that the object shape can be reflected in its signature. From figure 8.5, we observe that similarly shaped objects also show visual similarities in their measured hydrodynamic signatures, such as between the barrel and dumbbell shape and between the two capsule objects.

The hydrodynamic signature of the towed ball (Re ≈ 210000) closely resembles that of a modeled sphere in an inviscid fluid. This is remarkable, since the towed object generates a flow regime where vortices occur which would normally introduce chaos. While the vortices are still present in the data, as evidenced by the variation (percentiles, fills) in figure 8.5, combining these measurements produced a stable median signature.

Although the temporal hydrodynamic signatures are useful for visual inspection, the method of time-shifting signals is not well suited for classification, as it requires a constant speed and completed runs. To introduce more flexibility with respect to these constraints, we used a different processing method for the classification process.

8.5.2 *Feature and window based classification*

The approach of selecting feature subsets allowed optimizing the classifier, whereas the aggregating window approach made the method location-invariant and flexible towards measurement duration.

From the full list of selected features for each subset (table S2), some trends emerge. In general, the lower frequency bands (< 4 Hz) are selected more often than the higher frequency bands. There is no significant difference between the amount of selected x versus y dimension features.

The distribution features, kurtosis and skewness, are furthermore often chosen as informative during the feature selection process. This might be due to the low frequency shedding characteristics, which are less visible in the average hydrodynamic signatures (figure 8.5) given their quasi-random nature, but are more visible in individual temporal velocity profiles (figure 8.4).

The best performing feature subsets were the handpicked set containing both x and y frequency bands (98.6 ± 0.99 %), and the set of 17 features found via sequential backward search [114] (SBS, 95.8 ± 0.61 %). Another feature subset that stands out is the one found through FCBF [129] (94.6 ± 1.47 %), which only selects 8 features to achieve this classification performance. This illustrates that selecting informative features can be equally or more important compared to optimizing the classifier.

There are however some situations in which the classifier makes mistakes based on information from single four second time windows. From figure 8.7A it is clear that there is some confusion between the barrel, ball, and dumbbell shapes. This has likely to do with that these objects were measured from two distances, where objects further away produce more generic velocity profiles, as evidenced by the constructed hydrodynamic signatures and as expected from literature [19, 106]. For the capsule objects that were measured at a single distance, we also see some confusion, albeit less.

When we apply a majority vote per run, only few mistakes remain as indicated in figure 8.7B, regardless of the object's distance. The classification performance thus improves from mounting evidence, effectively averaging the prediction vector over sequential time windows in a run. The only mistakes that remain with a *per run* classifier (figure 8.7B) is

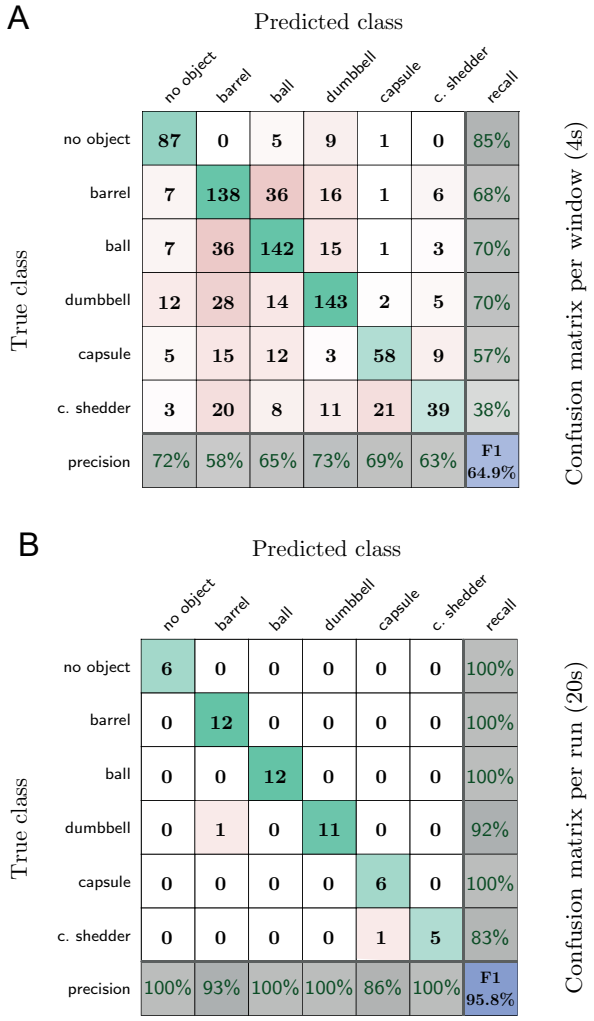


Figure 8.7: Confusion matrices for classification using the SBS feature set. (A) shows the predictions *per window*, while (B) shows the predictions after taking the *per run* approach.

that a dumbbell is mistaken for a barrel in one instance, and the capsule with a shedder is mistaken for a capsule without a shedder in another. Given the similarities between these shapes, these wrong predictions can still be considered informative.

8.5.3 *2D sensing*

The hydrodynamic signatures demonstrate the complementary advantage of 2D sensing; at zero crossings or low magnitudes in one sensing dimension, the other dimension often shows a local extrema when an object is nearby. This complementary effect is also visible from the final performance, where the best performance was achieved when combining both sensing dimensions.

Via the hand-picked feature subsets (figure 8.6), we find that the novel y component is more informative for discerning between objects. Combining the two measuring dimensions improves classification even further, which also holds for object localization [121].

8.5.4 *Future research*

Since the found feature subsets are tailored to discriminate between the considered object shapes, some features that were not chosen may still prove instrumental in discerning other shapes that may be added to a future data set or experiment.

For expansion on this research, and to further attest the utility and robustness of shape identification via hydrodynamic imaging, it would be interesting to combine the two current strategies for measuring the hydrodynamic effects of shapes. Both a stationary object placed upstream in a constant speed fluid flow, as well as an object moving with constant speed in still water, selectively consider one aspect of fluid flow sensing. Both the properties in Kármán vortex streets from literature and temporal hydrodynamic signatures presented here contain information about the object. As a first step, one could include an external flow source during measurements on moving objects, making for more realistic conditions.

Although the objects were moved at a single constant speed and at a constant distance, our classification pipeline does allow for deviations from this scheme. Due to the nature of the near-field hydrodynamic imaging, the objects should be close to the array, but not necessarily at a constant distance or for a fixed duration. Our processing pipeline ensures that the signals can be of arbitrary length and from any of the sensors from the array, and still produce a correct classification at a 4× chance rate on a single four second time window, as indicated by our *per window* scores (figure 8.7A). Taking a four second buffer into account, this pipeline could therefore be potentially be implemented online to provide online shape classification.

8.6 CONCLUSION

By up scaling the artificial lateral line (ALL), we have demonstrated that near-field hydrodynamic imaging can be used on a supra-biological scale. When deployed as a

static array, the system can be used for near-field object tracking and, as demonstrated here, object identification.

Furthermore we have not only shown that 2D sensing is beneficial with respect to shape identification, but also that the novel orthogonal velocity component bears more information than the traditional parallel velocity component, as used by fish and most 1D ALL applications.

This demonstrates that this 2D passive technology is especially suited to identify moving objects via their hydrodynamic interactions, and suggests that it can be used for object detection, object identification, and collision avoidance in dark and murky environments.

ACKNOWLEDGEMENTS

We thank Adrian Dzipalski, Steven Warmelink, Sybren Römer, and the Hanze Institute of Sport Studies for their contribution in facilitating the study featured in this study.

SUPPLEMENTARY INFORMATION

This section provides supplementary information for this study in the form of:

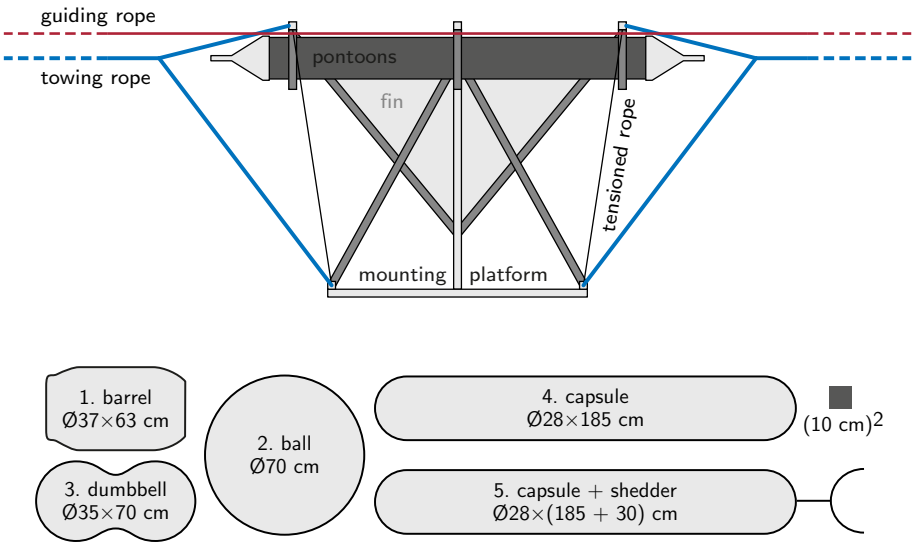


Figure 8.S1: Schematic side view (to scale) of the towing platform and the shapes used during the experiments.

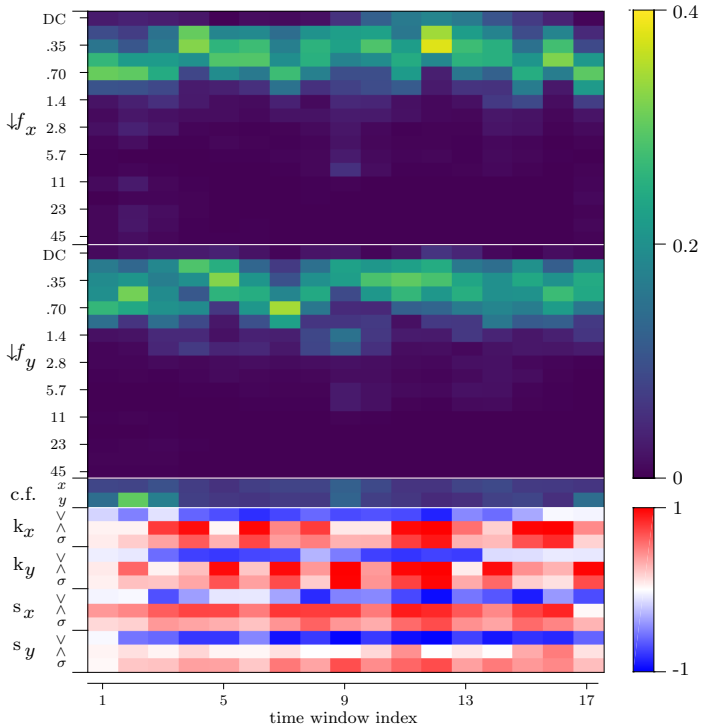


Figure 8.S2: Detailed example visualization of the feature space in figure 8.4. The feature values are represented in a color space indicated on the color bar on the right hand side. First we show the relative power of the normalized frequency bands. Second is the normalized centroid frequency (`c.f.`), and finally the kurtosis and skewness features listed in table 8.2.

Table 8.S1: List of features subsets, the number of features, and the 3-fold performance on the test set per run. The F1-score of the best performing handpicked and algorithmically determined subset is indicated in bold. The feature selection methods with a $*$ make use of a discretized version of the features. The shorthand symbols (minima, maxima, standard deviation) for kurtosis and skewness coincide with those of table 8.2.

SUBSET	x F-BANDS (Hz)	y F-BANDS (Hz)	C.F.	KURTOSIS AND SKEWNESS	#	F1-SCORE $\pm$ σ
all	DC to 45	DC to 45	x, y	all	48	96.3 ± 1.86 %
all f-bands	DC to 45	DC to 45	-	-	34	**98.6 ± 0.99 %**
x f-bands	DC to 45	-	-	-	17	75.7 ± 3.32 %
y f-bands	-	DC to 45	-	-	17	87.6 ± 3.41 %
c.f. & k+s	-	-	x, y	all	14	90.8 ± 1.88 %
Lasso	0.5, 1.4, 2.0, 2.8, 4.0, 5.7	0, 0.7, 1, 2.0, 11	-	$k_{x\vee}$, $k_{x\sigma}$, $s_{x\vee}$, $s_{x\sigma}$, $s_{y\wedge}$	16	85.7 ± 0.98 %
reliefF [102]	0.25	0.25, 0.35	-	all except ($k_{x\sigma}$, $s_{x\sigma}$)	13	78.6 ± 5.95 %
mRMR* [55]	-	0, 0.25, 4.0, 8.0	x	$s_{y\wedge}$, $s_{y\sigma}$	7	88.3 ± 1.35 %
FCBF* [129]	1.0	0.25, 4.0	x	$k_{x\vee}$, $k_{y\vee}$, $s_{x\vee}$, $s_{y\wedge}$	8	94.6 ± 1.47 %
cmRMR [37]	0, 1.0, 2.8, 4.0, 2.0, 16, 45	0, 0.7, 1.4, 11, 45	y	$s_{x\vee}$, $s_{y\sigma}$	15	91.0 ± 2.55 %
SBS [114]	0.25, 0.35, 0.5, 1.0, 23	0.25, 0.35, 1.0, 4.0	x	$k_{x\vee}$, $k_{x\sigma}$, $s_{x\vee}$, $s_{x\wedge}$, $s_{x\sigma}$, $s_{y\wedge}$, $s_{y\sigma}$	17	**95.8 ± 0.61 %**
SFS [114]	0, 0.25, 0.50, 23	0.35, 1.0, 2.0, 5.7	-	$k_{x\vee}$, $k_{y\vee}$, $s_{x\sigma}$, $s_{y\vee}$, $s_{y\wedge}$, $s_{y\sigma}$	14	94.6 ± 3.38 %

Table 8.S2: The 3-fold performance (F1-score ± σ) for each of the feature subsets for both training and testing sets. The # symbol denotes the number of features in each set. The *per window* scores denote classification based on a single four second window, the *per run* scores aggregate classifier output over sequential windows from a 20 second run.

SUBSET	#	TRAIN PER WINDOW	TEST PER WINDOW	TRAIN PER RUN	TEST PER RUN
all	48	74.8 ± 0.59	64.8 ± 1.23	100 ± 0	96.3 ± 1.86 %
all f-bands	34	69.8 ± 1.77	61.9 ± 1.38	98.6 ± 0.77	98.6 ± 0.99 %
x f-bands	17	58.4 ± 0.75	51.7 ± 1.06	80.0 ± 1.93	75.7 ± 3.32 %
y f-bands	17	58.0 ± 1.38	51.3 ± 1.02	95.0 ± 1.72	87.6 ± 3.41 %
c.f. & k,s	14	60.0 ± 0.56	52.6 ± 1.42	95.2 ± 0.88	90.8 ± 1.88 %
Lasso	16	63.5 ± 0.82	55.0 ± 0.69	96.4 ± 0.51	85.7 ± 0.98 %
reliefF	13	59.5 ± 1.45	51.8 ± 1.18	90.6 ± 2.85	78.6 ± 5.95 %
mRMR*	7	66.4 ± 0.88	62.6 ± 1.07	93.8 ± 1.05	88.3 ± 1.35 %
FCBF*	8	66.7 ± 0.87	60.9 ± 1.08	98.0 ± 0.72	94.6 ± 1.47 %
cmRMR	15	65.7 ± 1.34	60.4 ± 0.65	95.2 ± 1.83	91.0 ± 2.55 %
SBS	17	71.7 ± 1.04	64.1 ± 0.79	100 ± 0	95.8 ± 0.61 %
SFS	14	69.8 ± 0.59	62.6 ± 0.50	97.1 ± 0.91	94.6 ± 3.38 %

Part V

EPILOGUE

9 SUMMARY AND DISCUSSION

The four parts of this thesis combined a biophysical approach to artificial lateral line (ALL) sensing and a practical approach to AI-based techniques, in particular artificial neural networks, to perform what is known as hydrodynamic imaging. Via several simulation studies and several experiments ranging in scale from biomimetic to supra-biological, hydrodynamic imaging was demonstrated via localizing both vibrating and moving objects, as well as identifying an objects shape.

Many of these hydrodynamic imaging tasks have been demonstrated by others in the field, see [67] for a recent review. These demonstrations often employ 1D-sensitive sensors or use alternating sensor placements to capture more detail of the local hydrodynamic environment. Closer to the topic of this thesis, some recent examples describe multi-layer perceptron neural networks that perform object localization [4, 131], a two-class identification of objects [40], or a three-class identification of hydrodynamic environment [101]. The research featured in this thesis expanded on the current relevant literature in three main themes.

9.1 SCALABILITY

The sensing principle detailed in chapter 2, as well as the experiments in chapters 6, 7, and 8 have demonstrated that hydrodynamic imaging can be scaled up considerably. See also table 9.1 on the next page for an overview.

9.1.1 *Scalable technology*

The sensor design as presented in chapter 2 allows optimizing its sensitivity in terms of adjusting the frequency response. While the sensitivity-bandwidth product remains constant with any sensor design variation, one can be improved at the cost of the other. This allows the sensor design to be tuned and optimized for different flow regimes by varying the sensor dimensions and adjusting the geometry of the internal structure.

In addition, the all-optical technology enables large-scale deployments. The sensors make use of standard communication optical fibers and only assume a fiber connection to a remote laser-interrogator machine. This means that no electric or metal components need to be close to the sensor array or submerged, which is an advantage over (piezo)electric based sensing solutions, especially in a marine environment. Furthermore, compared to electrical signals, the loss of optical signals are several orders of

magnitude lower, allowing remote connected flow sensing on a distance over several kilometers.

9.1.2 *Hydrodynamic imaging at different scales*

Chapters 6 and 7 showed that it is feasible to scale up the task of object localization. Using a 16 cm relatively sparse array of four sensors in a small-scale setup, we demonstrated that we can localize a dipole object average error of 1.3 cm and moving objects with an error of 3.3 cm in an area of 36 by 11 cm. In chapter 7 we scale this situation up to an array of 45 cm with eight sensors to localize a moving object with an error of 0.72 cm in an area of 62 cm by 9.5 cm.

While the width of the detectable area for object localization increases from chapter 6 to 7, the detectable distance remains similar. This is ultimately a limit of the hydrodynamic signals that underlie hydrodynamic imaging. Hydrodynamic signals attenuate with a factor 1 over the distance cubed (d^{-3}), whereas for sonic signals as used in e.g. sonar, this attenuation factor is per distance squared (d^{-2}). This further emphasizes that hydrodynamic imaging is a near-field modality, but can be applied on a larger scale, if the object itself or its speed is large enough.

The larger scale in terms of object scale and object speed was demonstrated in chapter 8. Here, an array spanning 3.5 m is used to identify five different objects which move along the array with a constant speed of 0.3 m/s. It is obvious from the observed signals that the location of the object along the array can be easily retrieved. Shape recognition based on such signals has not been demonstrated on this scale before; the location-invariant classifier had a performance of of 98% (F1-score) for discriminating between the five object shapes.

9.1.3 *Suitability of scaling up hydrodynamic imaging*

The all-optical flow sensing system has allowed us to create large-scale flow sensor arrays, compared to the biological dimensions on fish. In addition, it allows sensing at more remote locations, without the use of electronics or electricity. While in chapter 8 the

Table 9.1: Overview of the tasks, amount (#) of sensors, and sizes as used in each chapter (§) that describes experiments.

§	TASK	# SENSORS	ARRAY LENGTH	DETECTION AREA
6	location	4	16 cm	36 × 11 cm
7	location	8	45 cm	62 × 9.5 cm
8	shape	8	350 cm	~ 600 × 100 cm

distance between the sensors and the measurement station was 15 meters, this can easily be extended using readily available optical fiber hardware. Furthermore, the sensors can be measured in parallel, by multiplexing the optical signals to a single optical fiber. This allows synchronous measurements of large-scale spatial velocity profiles. The technical solution is therefore especially suited for hydrodynamic imaging.

Via several experiments, we have increased the scale on which hydrodynamic imaging is performed from the centimeter scale to several meters. Since hydrodynamic imaging is a passive, near-field modality, the detectable distance of an object is more dependent more on the object itself than the sensor. The hydrodynamic signal is a result of water displacement, so: the bigger and faster the object, the stronger the hydrodynamic signal.

In conclusion, while it is hard to increase the detection range of a single sensor, by using larger arrays and more sensors, the area in which an object can be detected using hydrodynamic imaging can be increased considerably.

9.2 NEURAL-NETWORK BASED SIGNAL PROCESSING

Several variations of neural networks have been featured throughout the thesis demonstrating that, depending on the task at hand, there are performance gains by selecting the right type of carefully optimized neural network. A complete overview of which type of neural networks were used for a given task is listed here as table 9.2.

Table 9.2: Overview of sensor types and tasks using neural-network based signal processing. For each of the chapters (§), we indicate whether it is a simulation study (sim) or experiment (exp), and also indicate the type of data processing used.

§	TASK	SENSOR	SOURCE	NN TYPE	DATA
3	location	1D sim	moving	MLP, ELM, ESN	spatial
4	likelihood	1D sim	moving	CNN	both
5	location & direction	1D sim & 2D sim	moving & vibrating	ELM	both
6	location	2D exp	moving & vibrating	ELM	both
7	location	2D exp	moving	OS-ELM, LSTM	both
8	shape	2D exp	moving	ELM	temporal

Overall, four conceptually different tasks have been performed: locating a stationary vibrating object, locating a moving object, determining the likelihood of an object present in an area, and determining the shape of a moving object. These are different in the way temporal and spatial information from a velocity pattern is treated.

9.2.1 *Focusing on the spatial dimension*

This type of processing is useful for localizing a object that is stationary, for instance a vibrating engine or in our case a dipole source. In this case, we mainly consider the spatial velocity profile information; the temporal information is averaged to a single spatial profile to be processed further. Effectively, the temporal component of a measurement is only used for signal to noise reduction. This type of processing occurred in chapters 3, 5, and 6.

In the simulated dipole localization study of chapter 5, we substituted the effect of averaging over time by varying the level of added simulated sensing noise. The selected noise levels were chosen relative to the weakest observed signal without noise present. The results showed that it took a noise level two orders of magnitude higher than this weakest observed signal to reduce the performance of an ELM from optimal to random chance. This was for determining both the location and vibration direction of the object.

For the experiments described in chapter 6, we only had access to a limited data set to demonstrate whether the ELM is capable of dipole source localization. The reconstruction of the dipole's location using just four sensors resulted in an average distance error of 1.3 cm for the unseen test set, versus 0.45 cm on the training data. The ELM thus proved to be capable to localize a dipole source in an experimental setting.

While in chapter 3 we considered a simulated moving object, the ELM and MLP neural networks in this chapter only used a single time step as input. These two networks only focused on spatial information. As in chapter 5, different levels of noise were added to assert performance robustness and the effect of averaging over time. The ESN was also considered since it makes use of past inputs, but ultimately only provided an improvement at higher noise levels. Nevertheless, we hypothesized here that recurrent and temporal information may still be relevant for locating moving objects.

9.2.2 *Combining temporal and spatial information*

This type of processing has been shown useful in most cases for localizing moving objects. In this case, the temporal dimension is either explicitly considered via using several sequential velocity profiles as inputs, or the neural networks has internal dynamics that implicitly use temporal information.

As mentioned in the previous section on chapter 3, the temporal component is implicitly available to the ESN neural network. When we simulated some added measurement noise, the recurrent connections in the ESN did help to improve the performance for

localizing a simulated moving object. This suggested that temporal information may be informative for the experiments described in parts III and IV.

For localizing a moving object within a medium-scale experiment (chapter 6), temporal information was explicitly added to the neural network as concatenated input. In this experiment, we limited the duration of considered history to 500 ms to enable timely output; the amount of history samples was varied from none (only the current sample) to the past 45 inputs. Consistently, the cross-validation performance correlated more with the total duration of history, compared to the amount of past samples. The duration of considered input history was therefore more important than the resolution of the considered input history.

This beneficial effect of recurrent connection for localizing moving objects was further demonstrated in chapter 7. Here, both the LSTM and ELM were using explicit history, in the sense that a time window of 30 samples was used as an input. The LSTM also uses recurrency via its memory blocks; this allowed the LSTM to outperform the ELM architecture significantly for this task, and proved to be more consistent in its performance as well. We therefore conclude that recurrency can be a beneficial aspect in signal processing for localizing a moving source.

There are of course limits to the length of history that can be deemed relevant or helpful for neural networks, depending on the situation. If the object has short motions or traces an erratic path, we lose long-term temporal correlation. An example of this can be seen in chapter 4, where a CNN was used to determine the likelihood of a simulated object present in a near-field 2D map. We observed that considering only the current time step was optimal compared to considering up to four past inputs. This may have been a result of the size and nature of the simulated data set, where objects appeared and disappeared randomly. This suggests, as one would expect, that learning patterns through time will require more training data compared to only learning spatial patterns to result in a boost in performance. When the available data set is small or the object motion less predictable, temporal information may therefore not be beneficial for hydrodynamic imaging.

9.2.3 *Focusing on the temporal dimension*

In the final experiment we identified the object shape, regardless of relative position and direction of motion. In this case only the temporal velocity profile information is considered; the spatial information was combined or averaged to a single feature vector.

In this study, we preprocessed the data in time windows of fixed time duration, where some spatial information was still implicitly available via the implementation of the kurtosis and skewness feature values. Given the limited size of the data set, it is likely that adding more (spatial) features to the classifier would not have boosted the performance. Especially since the feature selection process revealed that smaller subsets of features lead to better performance.

There are however other means to process the data from this set of measurements. Using correlation between the temporal signals of each sensor array, it is trivial to deduce the speed of the object. Since the speed of each object was identical, this deduced speed of the object is not informative, but may very well become so for other applications where objects of interest have different typical speeds.

9.2.4 *Suitability of neural networks for hydrodynamic imaging*

Overall, we find that considering both the temporal and spatial dimension is optimal in most cases; either as a method to improve on the signal to noise level, or to provide additional meaningful inputs to a signal processing pipeline.

We have tried and compared different types of neural networks, with the default being a light-weight reservoir neural network, the ELM. Other network architectures that were considered are the: MLP, ESN, LSTM, and CNN. These neural network frameworks considered have all shown to be effective and capable to perform a task for hydrodynamic imaging.

The type of network and type of processing that is best suited for a given hydrodynamic imaging task depends on the nature and availability of the data. The amount of useful inputs, spatial or temporal, does not only depend on the intrinsic usefulness of the feature itself. The more data is available, the more likely it is that patterns can be extracted from the data, be it spatial, temporal, or otherwise.

In conclusion, neural networks, when carefully optimized to avoid overfitting, have shown to be very well suited to perform hydrodynamic imaging.

9.3 2D SENSING

We have shown that 2D-sensitive sensors were able to demonstrate all given tasks, and furthermore demonstrated that 2D sensing has a complementary effect on top of being a noise reduction effect.

9.3.1 *Demonstration of 2D sensing*

The first direct comparison between 1D-sensitive sensing and 2D-sensitive sensing in simulation is featured in chapter 5. In this chapter, we first point toward the possible complementary effect of 2D sensing. Based on the modeled velocity profiles we observe that, for a spherical object, the two sensing dimensions complement each other for spatial characteristics; where one dimension shows a zero crossing in the spatial velocity profile, the other shows a maximum or minimum. This entails that at these informative spatial points, there is a considerable gain with respect to the signal to noise ratio.

For the simulation study in chapter 5, the setup consisting of 2D-sensitive sensors lead to half the error compared to their 1D counterpart under noiseless conditions. This relative improvement declines at increased levels of noise, but is still present in the case for determining the location of a vibrating or moving object. Further analysis on the spatial distribution of errors revealed that, especially in the areas near and next to the array, the 2D-sensitive sensors produce a considerable boost in performance.

The experiments featured in chapter 6, 7, and 8 all use 2D-sensitive sensors where no direct comparison is made between using just one of the velocity components or both. However, through further analysis of the measured velocity profiles, we have made observations that support the notion from chapter 5, which point towards the complementary effect of 2D sensing.

In chapter 6, we visualized 2D velocity patterns per sensor. This showed that the zero-crossings do show up in these spatial profiles and that they furthermore adhere to the modeled velocity profiles. In chapter 8, we visualized *temporal* velocity patterns per object. While it does not guarantee that the information was used as such by the processing pipeline, these temporal profiles showed the same complementary aspect for some of the shapes considered.

Finally, via the performance of different subsets of features in the shape classification study from chapter 8, we have a clear indication that combining the two dimensions is the optimal case for shape recognition. Furthermore, we found that the novel, orthogonal sensing dimension is more informative than the traditional parallel component for discerning the shapes used in this experiment.

9.3.2 *Suitability of 2D sensing for hydrodynamic imaging*

In conclusion, for both localizing an object and determining its shape, we find that combining the two sensing dimensions in a single sensor is very well suited for hydrodynamic imaging.

9.4 FUTURE RESEARCH

Past chapters have listed directions for future research; some of these suggestions have already been executed in subsequent chapters. We here list several future directions and considerations that were not performed in the previous chapters.

9.4.1 *Further development*

The sensor as presented in chapter 2 was inspired by the fluid-interaction model of a subdermal fish canal neuromast. Both in biophysical experiments and biomimetic artificial lateral lines, it has been shown that the canal provides an additional DC filtering component. While we have encased the sensors by protective tubes in chapters 7 and 8,

these caused no significant filtering to occur. For future implementations of all-optical flow sensing, it may be interesting to adapt the sensor housing to allow such filtering to occur, making the sensor less sensitive to DC flow and more sensitive towards measuring dynamic hydrodynamic signals.

Another alteration to the current setup that could prove to be an improvement is to consider different array geometries. While we only considered equidistant sensors on a line in this thesis, alternative sensor configurations could increase the area in which hydrodynamic imaging is possible. One could consider placing the sensors on the perimeter of an area of interest or in a grid formation within an area of interest. Such alternative configurations would reduce the ability to measure spatial velocity profiles in the traditional sense, but the information contained in the measurements from alternative geometry systems may be as informative as the traditional line.

9.4.2 *Further comparisons*

As mentioned in the introduction, there are two modes of passive hydrodynamic imaging. We mainly focused on the flow generated by a moving object in still water, but a stationary object upstream produces periodic wake shedding which also contains information. As stated in chapter 8, these two modalities are often studied separately in the literature. In the real world, it is more likely for both to occur simultaneously, with for instance a sailing vessel or a fish swimming upstream. Although it can be challenging to perform, it would be interesting to find an application that combines both modalities to perform hydrodynamic imaging. This would help bring hydrodynamic imaging from a demonstration to a real world application.

Finally, although we have mainly focused on neural network based signal processing in this thesis, there are several methods reported in the literature that also perform dipole source localization, such as template matching and beamforming. Given that the area of interest, data set size, sensor sensitivity, and sensor modalities differ between these studies, it is challenging to make a fair comparison between the performance of these signal processing methods. It would be useful to have an overview of these methods in a comparative study, further putting the achieved results from the thesis and the literature into context.

9.4.3 *Further applications*

With the current 2D-sensitive flow sensor array, there are additional phenomena that are interesting to measure and research.

As we deployed the sensor system in a swimming pool in chapter 8, it might be useful as a research tool for competitive swimming. We could for instance discern different swimming techniques and strokes or identify individual swimmers, using the same feature-based approach. In addition, one could also use the flow sensing array to

study the effectiveness of a swimming stroke by measuring the generated flow field. A 2D-sensitive flow sensing array may therefore provide a helpful tool in sports research.

Another area of research, the field of oceanography, may also benefit from the system featured in the thesis. One beneficial aspect of the system is that the all-optical sensor array system allows continuous measurements over large distances. Furthermore it enables measurements near the seabed floor, a challenging region for other flow measuring systems. In addition, the current sensing method is unaffected by murky waters, which is an advantage over sonar or vision based solutions.

Measuring the hydrodynamics of larger bodies of water could also have further applications for the renewable energy sector. One example of such an application is the harvesting of energy from tides. In these kinds of applications, it is important to have a clear picture of the hydrodynamic environment, before deploying for instance an underwater tidal turbine. The expected bulk flow rate is of course important for estimating the yield of the generator, but hydrodynamic vorticity and turbulence also affect the system as a whole and are relevant for its integrity.

Finally, 2D-sensitive flow sensors may also improve obstacle avoidance for e.g. underwater autonomous vehicles. Specifically for the system as presented in the thesis, mounting an all-optical system would require the development of an interrogator module on the platform itself. But other flow sensing or multi-modal solutions could also make use of 2D sensing.

9.4.4 *Other applications*

While we have discussed the results and novel contributions to the field, as well as possible tracks for future research in the field, there may be other applications beyond the scope of this thesis that could benefit from the work presented here.

When we set the main application of hydrodynamic imaging in bodies of water aside for a moment, the studies and experiments in this thesis contributed towards solving (parts of) the inverse problem. In abstract terms, we used an array of sensors to measure the effect of a stimulus to work out what the stimulus is. There are many more applications in which such an inverse problem needs to be solved, and in which neural-network based signal processing can provide, or already provide [7], a viable alternative.

One such application is the field of electroencephalography (EEG) brain imaging, where a range of electroreceptive sensors are positioned on the scalp and provide multi-channel data with a high temporal resolution. This multi-channel signal is then used to classify patterns or used to estimate the location of dipole sources that can explain the measured data. While beamforming and other covariance analyses are the default, many brain-computer interface (BCI) applications already make use of machine learning [83].

As is the case in fluid dynamics, there is another area of research where the inversion of forward model is either computationally intensive or an approximation: seismic

inversion. In this geophysics application, an array of sensors is used to measure the reflection of a controlled seismic stimulus to determine the composition of a volume of the earth's crust, usually for detecting oil and gas fields. This field is also starting to take inspiration from the fish lateral line [44], and employs neural-network based signal processing [14].

9.5 CONCLUSION

In conclusion, the novel contributions of this thesis to the field of artificial lateral line hydrodynamic imaging can be summarized among three themes: scalability, neural-network based signal processing, and 2D-sensing. We have shown that hydrodynamic imaging can be scaled up considerably using our technology, that neural networks are very well suited for hydrodynamic imaging, and that 2D sensing improves the performance overall.

While we focused here on determining the location, orientation, and shape of submerged objects at a distance, there are several future directions to pursue, both inside and outside the water that has inspired the development of this sensing solution.

BIBLIOGRAPHY

[1] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[2] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn. "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition." In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE. 2012, pp. 4277–4280. DOI: 10.1109/ICASSP.2012.6288864.

[3] A. T. Abdulsadda and X. Tan. "Underwater source localization using an IPMC-based artificial lateral line." In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 2719–2724. DOI: 10.1109/ICRA.2011.5980545.

[4] A. T. Abdulsadda and X. Tan. "An artificial lateral line system using IPMC sensor arrays." en. In: *International Journal of Smart and Nano Materials* 3.3 (2012), pp. 226–242. DOI: 10.1080/19475411.2011.650233.

[5] A. T. Abdulsadda and X. Tan. "Nonlinear estimation-based dipole source localization for artificial lateral line systems." In: *Bioinspiration & Biomimetics* 8.2 (2013), p. 026005. DOI: 10.1088/1748-3182/8/2/026005.

[6] A. T. Abdulsadda and X. Tan. "Underwater tracking of a moving dipole source using an artificial lateral line: algorithm and experimental validation with ionic polymer–metal composite flow sensors." In: *Smart Materials and Structures* 22.4 (2013). DOI: 10.1088/0964-1726/22/4/045010.

[7] J. Adler and O. Öktem. "Solving ill-posed inverse problems using iterative deep neural networks." In: *Inverse Problems* 33.12 (2017), p. 124007. ISSN: 0266-5611, 1361-6420. DOI: 10.1088/1361-6420/aa9581.

[8] A. Ahrari, H. Lei, M. A. Sharif, K. Deb, and X. Tan. "Reliable underwater dipole source characterization in 3D space by an optimally designed artificial lateral line system." In: *Bioinspiration & Biomimetics* 12.3 (2017), p. 036010. DOI: 10.1088/1748-3190/aa69a4.

[9] M. Alsalman, B. Colvert, and E. Kanso. "Training bioinspired sensors to classify flows." In: *Bioinspiration & Biomimetics* 14.1 (2018), p. 016009. ISSN: 1748-3190. DOI: 10.1088/1748-3190/aaef1d.

[10] F. M. Araujo, L. A. Ferreira, and J. L. Santos. "Simultaneous determination of curvature, plane of curvature, and temperature by use of a miniaturized sensing head based on fiber Bragg gratings." In: *Applied optics* 41.13 (2002), pp. 2401–2407. DOI: 10.1364/ao.41.002401.

[11] M. Asadnia, A. G. P. Kottapalli, Z. Shen, J. Miao, and M. Triantafyllou. "Flexible and surface-mountable piezoelectric sensor arrays for underwater sensing in marine vehicles." In: *IEEE Sensors Journal* 13.10 (2013), pp. 3918–3925. DOI: 10.1109/JSEN.2013.2259227.

[12] M. Asadnia, A. G. P. Kottapalli, J. Miao, M. E. Warkiani, and M. S. Triantafyllou. "Artificial fish skin of self-powered micro-electromechanical systems hair cells for sensing hydrodynamic flow phenomena." en. In: *Journal of The Royal Society Interface* 12.111 (2015), p. 20150322. DOI: 10.1098/rsif.2015.0322.

[13] M. Asadnia, A. G. P. Kottapalli, K. D. Karavitaki, M. E. Warkiani, J. Miao, D. P. Corey, and M. Triantafyllou. "From Biological Cilia to Artificial Flow Sensors: Biomimetic Soft Polymer Nanosensors with High Sensing Performance." In: *Scientific Reports* 6 (2016), p. 32955. DOI: 10.1038/srep32955.

[14] M. van der Baan and C. Jutten. "Neural networks in geophysical applications." en. In: *Geophysics* 65.4 (July 2000), pp. 1032–1047. ISSN: 0016-8033, 1942-2156. DOI: 10.1190/1.1444797.

[15] Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives." In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50.

[16] P. P. Benham, C. Armstrong, and R. J. Crawford. *Mechanics of engineering materials*. Addison-Wesley, 1996.

[17] J. Bergstra and Y. Bengio. "Random search for hyper-parameter optimization." In: *Journal of Machine Learning Research* 13.Feb (2012), pp. 281–305. URL: http://www.jmlr.org/papers/v13/bergstra12a.html.

[18] H. Bleckmann, J. Mogdans, J. Engelmann, S. Kröther, and W. Hanke. "Wie fische wasser fühlen: das seitenliniensystem." de. In: *Biologie in unserer Zeit* 34.6 (2004), pp. 358–365. DOI: 10.1002/biuz.200410266.

[19] R. Bouffanais, G. D. Weymouth, and D. K. P. Yue. "Hydrodynamic object recognition using pressure sensing." en. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 467.2125 (2011), pp. 19–38. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2010.0095.

[20] L. H. Boulogne, B. J. Wolf, M. A. Wiering, and S. M. van Netten. "Performance of neural networks for localizing moving objects with an artificial lateral line." In: *Bioinspiration & Biomimetics* 12.5 (2017), p. 056009. DOI: 10.1088/1748-3190/aa7fcb.

[21] B. P. Chagnaud and S. Coombs. "Information Encoding and Processing by the Peripheral Lateral Line System." In: *The Lateral Line System*. Ed. by S. Coombs, H. Bleckmann, R. R. Fay, and A. N. Popper. Vol. 48. New York, NY: Springer New York, 2013, pp. 151–194. ISBN: 978-1-4614-8850-7 978-1-4614-8851-4. DOI: 10.1007/2506_2013_15.

[22] L. D. Chambers, O. Akanyeti, R. Venturelli, J. Je ov, J. Brown, M. Kruusmaa, P. Fiorini, and W. M. Megill. "A fish perspective: detecting flow features while moving using an artificial lateral line in steady and unsteady flow." en. In: *Journal of The Royal Society Interface* 11.99 (2014), pp. 20140467–20140467. ISSN: 1742-5689, 1742-5662. DOI: 10.1098/rsif.2014.0467.

[23] X. Chen, G. Zhu, X. Yang, D. L. S. Hung, and X. Tan. "model-based estimation of flow characteristics using an ionic polymer metal composite beam." In: *IEEE/ASME Transactions on Mechatronics* 18.3 (2013), pp. 932–943. DOI: 10.1109/TMECH.2012.2194300.

[24] F. Chollet et al. *Keras*. https://keras.io. 2015.

[25] B. Colvert, M. Alsalman, and E. Kanso. "Classifying vortex wakes using neural networks." In: *Bioinspiration & Biomimetics* 13.2 (2018), p. 025003. ISSN: 1748-3190. DOI: 10.1088/1748-3190/aaa787.

[26] S. Coombs and R. A. Conley. "Dipole source localization by mottled sculpin. I. Approach strategies." In: *Journal of Comparative Physiology A* 180.4 (1997), pp. 387–399. DOI: 10.1007/s003590050057.

[27] S. Coombs, J. Janssen, and J. F. Webb. "Diversity of lateral line systems: evolutionary and functional considerations." In: *Sensory biology of aquatic animals*. Springer, 1988, pp. 553–593. DOI: 10.1007/978-1-4612-3714-3_22.

[28] S. Coombs and J. C. Montgomery. "The enigmatic lateral line system." In: *Comparative hearing: fish and amphibians*. Ed. by R. R. Fay, A. N. Popper, R. R. Fay, and A. N. Popper. Vol. 11. New York, NY: Springer New York, 1999, pp. 319–362. DOI: 10.1007/978-1-4612-0533-3_8.

[29] S. Coombs and J. C. Montgomery. "Comparing Octavolateralis Sensory Systems: What Can We Learn?" en. In: *Electroreception*. Ed. by T. H. Bullock, C. D. Hopkins, A. N. Popper, and R. R. Fay. Vol. 21. Springer New York, 2005, pp. 318–359. ISBN: 978-0-387-23192-1. DOI: 10.1007/0-387-28275-0_12.

[30] S. Coombs and S. van Netten. "The hydrodynamics and structural mechanics of the lateral line system." In: *Fish physiology* 23 (2005), pp. 103–139. DOI: 10.1016/S1546-5098(05)23004-2.

[31] B. Curcic-Blake and S. M. van Netten. "Source location encoding in the fish lateral line canal." en. In: *Journal of Experimental Biology* 209.8 (2006), pp. 1548–1559. DOI: 10.1242/jeb.02140.

[32] A. Dagamseh, R. Wiegerink, T. Lammerink, and G. Krijnen. "Imaging dipole flow sources using an artificial lateral-line system made of biomimetic hair flow sensors." en. In: *Journal of The Royal Society Interface* 10.83 (2013), p. 20130162. ISSN: 1742-5689, 1742-5662. DOI: 10.1098/rsif.2013.0162. (Visited on 06/07/2016).

[33] A. Dagamseh, T. Lammerink, M. Kolster, C. Bruinink, R. Wiegerink, and G. Krijnen. "Dipole-source localization using biomimetic flow-sensor arrays positioned as lateral-line system." en. In: *Sensors and Actuators A: Physical* 162.2 (2010), pp. 355–360. DOI: 10.1016/j.sna.2010.02.016.

[34] A. Dagamseh, R. Wiegerink, T. Lammerink, and G. Krijnen. "Artificial lateral-line system for imaging dipole sources using beamforming techniques." In: *Procedia Engineering* 25 (2011), pp. 779–782. DOI: 10.1016/j.proeng.2011.12.191.

[35] L. DeVries, F. D. Lagor, H. Lei, X. Tan, and D. A. Paley. "Distributed flow estimation and closed-loop control of an underwater vehicle with a multi-modal artificial lateral line." In: *Bioinspiration & Biomimetics* 10.2 (2015), p. 025002. ISSN: 1748-3190. DOI: 10.1088/1748-3190/10/2/025002.

[36] S. Dijkgraaf. "The functioning and significance of the lateral-line organs." In: *Biological Reviews* 38.1 (1963), pp. 51–105. DOI: 10.1111/j.1469-185X.1963.tb00654.x.

[37] C. Ding and H. Peng. "Minimum redundancy feature selection from microarray gene expression data." In: *Journal of bioinformatics and computational biology* 3.2 (2005), pp. 185–205.

[38] Z. Fan, J. Chen, J. Zou, D. Bullen, C. Liu, and F. Delcomyn. "Design and fabrication of artificial lateral line flow sensors." In: *Journal of Micromechanics and Microengineering* 12.5 (2002), p. 655. DOI: 10.1088/0960-1317/12/5/322.

[39] A. Fender et al. "Two-axis accelerometer based on multicore fibre Bragg gratings." In: *Proceedings Third European Workshop on Optical Fibre Sensors*. 2007, pp. 6619–4. DOI: 10.1117/12.738411.

[40] V. I. Fernandez, A. Maertens, F. M. Yaul, J. Dahl, J. H. Lang, and M. S. Triantafyllou. "Lateral-line-inspired sensor arrays for navigation and object identification." In: *Marine Technology Society Journal* 45.4 (2011), pp. 130–146.

[41] Å. Flock and J. Wersäll. "A study of the orientation of the sensory hairs of the receptor cells in the lateral line organ of fish, with special reference to the function of the receptors." In: *The Journal of Cell Biology* 15.1 (1962), pp. 19–27. DOI: 10.1083/jcb.15.1.19.

[42] G. M. H. Flockhart, W. N. MacPherson, J. S. Barton, J. D. C. Jones, L. Zhang, and I. Bennion. "Two-axis bend measurement with Bragg gratings in multicore optical fiber." en. In: *Optics Letters* 28.6 (2003), p. 387. DOI: 10.1364/OL.28.000387.

[43] J.-M. P. Franosch, A. B. Sichert, M. D. Suttner, and J. L. van Hemmen. "Estimating position and velocity of a submerged moving object by the clawed frog Xenopus and by fish—a cybernetic approach." In: *Biological cybernetics* 93.4 (2005), pp. 231–238. DOI: 10.1007/s00422-005-0005-0.

[44] F. W. d. Freitas Silva, S. L. E. F. da Silva, M. V. C. Henriques, and G. Corso. "Using fish lateral line sensing to improve seismic acquisition and processing." en. In: *PLOS ONE* 14.4 (Apr. 2019). Ed. by H. V. Ribeiro, e0213847. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0213847.

[45] X. Gao and T. Zhang. "Unsupervised learning to detect loops using deep neural networks for visual SLAM system." In: *Autonomous Robots* (2015), pp. 1–18. ISSN: 1573-7527. DOI: 10.1007/s10514-015-9516-2.

[46] A. Ghysen and C. Dambly-Chaudiere. "The lateral line microcosmos." en. In: *Genes & Development* 21.17 (2007), pp. 2118–2130. DOI: 10.1101/gad.1568407.

[47] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. Vol. 9. 2010, pp. 249–256.

[48] X. Glorot, A. Bordes, and Y. Bengio. "Deep Sparse Rectifier Neural Networks." In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by G. Gordon, D. Dunson, and M. Dudík. Vol. 15. Proceedings of Machine Learning Research. PMLR, 2011, pp. 315–323. URL: http://proceedings.mlr.press/v15/glorot11a.html.

[49] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[50] J. Goulet, J. Engelmann, B. P. Chagnaud, J.-M. P. Franosch, M. D. Suttner, and J. L. van Hemmen. "Object localization through the lateral line system of fish: theory and experiment." en. In: *Journal of Comparative Physiology A* 194.1 (2008), pp. 1–17. DOI: 10.1007/s00359-007-0275-1.

[51] A. Graves. "Supervised sequence labelling." In: *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 5–13. DOI: 10.1007/978-3-642-24797-2_2.

[52] A. Graves, A.-r. Mohamed, and G. Hinton. "Speech recognition with deep recurrent neural networks." In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE. 2013, pp. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.

[53] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. "LSTM: A search space odyssey." In: *IEEE transactions on neural networks and learning systems* 28.10 (2017), pp. 2222–2232. DOI: 10.1109/TNNLS.2016.2582924.

[54] I. Guyon and A. Elisseeff. "An introduction to variable and feature selection." In: *Journal of machine learning research* 3 (2003), pp. 1157–1182.

[55] Hanchuan Peng, Fuhui Long, and C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." en. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1226–1238. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.159.

[56] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." In: *arXiv preprint arXiv:1512.03385* (2015).

[57] G. Heinzel, A. Rüdiger, and R. Schilling. "Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new at-top windows." In: (2002). DOI: 11858/00-001M-0000-0013-557A-5.

[58] H. Herzog, S. Steltenkamp, A. Klein, S. Tätzner, E. Schulze, and H. Bleckmann. "Micro-Machined Flow Sensors Mimicking Lateral Line Canal Neuromasts." en. In: *Micromachines* 6.8 (2015), pp. 1189–1212. ISSN: 2072-666X. DOI: 10.3390/mi6081189.

[59] K. O. Hill and G. Meltz. "Fiber Bragg grating technology fundamentals and overview." In: *Journal of lightwave technology* 15.8 (1997), pp. 1263–1276. DOI: 10.1109/50.618320.

[60] S. Hochreiter and J. Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[61] D. Hoekstra and J. Janssen. "Non-visual feeding behavior of the mottled sculpin, Cottus bairdi, in Lake Michigan." In: *Environmental biology of fishes* 12.2 (1985), pp. 111–117. DOI: 10.1007/BF00002763.

[62] K. Hornik. "Approximation capabilities of multilayer feedforward networks." In: *Neural networks* 4.2 (1991), pp. 251–257. DOI: 10.1016/0893-6080(91)90009-T.

[63] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. "Extreme learning machine: a new learning scheme of feedforward neural networks." In: *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*. Vol. 2. IEEE. 2004, pp. 985–990. DOI: 10.1109/IJCNN.2004.1380068.

[64] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. "Extreme learning machine: theory and applications." In: *Neurocomputing* 70.1 (2006), pp. 489–501. DOI: 10.1016/j.neucom.2005.12.126.

[65] A. N. S. Institute. *Specification for octave-band and fractional-octave-band analog and digital filters*. 2004.

[66] H. Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*. German National Research Center for Information Technology, 2002.

[67] Y. Jiang, Z. Ma, and D. Zhang. "Flow field perception based on the fish lateral line system." en. In: *Bioinspiration & Biomimetics* 14.4 (2019), p. 041001. ISSN: 1748-3190. DOI: 10.1088/1748-3190/ab1a8d.

[68] A. J. Kalmijn. "Hydrodynamic and Acoustic Field Detection." In: *Sensory Biology of Aquatic Animals*. Ed. by J. Atema, R. R. Fay, A. N. Popper, and W. N. Tavolga. New York, NY: Springer New York, 1988, pp. 83–130. DOI: 10.1007/978-1-4612-3714-3_4.

[69] A. J. Kalmijn. "Hydrodynamic and acoustic field detection." In: *Sensory biology of aquatic animals*. Springer, 1988, pp. 83–130. DOI: 10.1007/978-1-4612-3714-3_4.

[70] A. T. Klein, F. Kaldenbach, A. Rüter, and H. Bleckmann. "What We Can Learn from Artificial Lateral Line Sensor Arrays." In: *The Effects of Noise on Aquatic Life II*. Ed. by A. N. Popper and A. Hawkins. Vol. 875. New York, NY: Springer New York, 2016, pp. 539–545. ISBN: 978-1-4939-2980-1 978-1-4939-2981-8.

[71] A. G. P. Kottapalli, M. Asadnia, J. Miao, and M. Triantafyllou. "Soft polymer membrane micro-sensor arrays inspired by the mechanosensory lateral line on the blind cavefish." en. In: *Journal of Intelligent Material Systems and Structures* 26.1 (2015), pp. 38–46. DOI: 10.1177/1045389X14521702.

[72] A. G. P. Kottapalli, M. Asadnia, H. Hans, J. M. Miao, and M. S. Triantafyllou. "Harbor seal inspired MEMS artificial micro-whisker sensor." In: *Micro Electro Mechanical Systems (MEMS), 2014 IEEE 27th International Conference on*. IEEE, 2014, pp. 741–744. DOI: 10.1109/MEMSYS.2014.6765747.

[73] M. Krieg, K. Nelson, and K. Mohseni. "Distributed sensing for fluid disturbance compensation and motion control of intelligent robots." en. In: *Nature Machine Intelligence* 1.5 (2019), pp. 216–224. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0044-1.

[74] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: vol. 60. 6. New York, NY, USA: ACM, 2017, pp. 84–90. DOI: 10.1145/3065386.

[75] A. Kroese and N. Schellart. "Velocity-and acceleration-sensitive units in the trunk lateral line of the trout." In: *Journal of Neurophysiology* 68.6 (1992), pp. 2212–2221. DOI: 10.1152/jn.1992.68.6.2212.

[76] M. Kruusmaa, P. Fiorini, W. Megill, M. de Vittorio, O. Akanyeti, F. Visentin, L. Chambers, H. El Daou, M.-C. Fiazza, J. Ježov, et al. "Filose for svenning: A flow sensing bioinspired robot." In: *IEEE Robotics & Automation Magazine* 21.3 (2014), pp. 51–62. DOI: 10.1109/MRA.2014.2322287.

[77] S. Lakkam, B. T. Balamurali, and R. Bouffanais. "Hydrodynamic object identification with artificial neural models." en. In: *Scientific Reports* 9.1 (2019). ISSN: 2045-2322. DOI: 10.1038/s41598-019-47747-8.

[78] H. Lamb. *Hydrodynamics*. Cambridge university press, 1932.

[79] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[80] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan. "A fast and accurate online sequential learning algorithm for feedforward networks." In: *IEEE Transactions on neural networks* 17.6 (2006), pp. 1411–1423. DOI: 10.1007/s10462-013-9405-z.

[81] G. Liu, A. Wang, X. Wang, and P. Liu. "A Review of artificial lateral line in sensor fabrication and bionic applications for robot fish." en. In: *Applied Bionics and Biomechanics* 2016 (2016), pp. 1–15. DOI: 10.1155/2016/4732703.

[82] G. Liu, M. Wang, A. Wang, S. Wang, T. Yang, R. Malekian, and Z. Li. "Research on Flow Field Perception Based on Artificial Lateral Line Sensor System." en. In: *Sensors* 18.3 (2018), p. 838. DOI: 10.3390/s18030838.

[83] F Lotte, L Bougrain, A Cichocki, M Clerc, M Congedo, A Rakotomamonjy, and F Yger. "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update." In: *Journal of Neural Engineering* 15.3 (2018), p. 031005. ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2552/aab2f2. (Visited on 11/14/2019).

[84] P. Lu and Q. Chen. "Fiber Bragg grating sensor for simultaneous measurement of flow rate and direction." In: *Measurement Science and Technology* 19.12 (2008), p. 125302. DOI: 10.1088/0957-0233/19/12/125302.

[85] A. M. Mathai, P. Moschopoulos, and G. Pederzoli. "Random points associated with rectangles." In: *Rendiconti del Circolo Matematico di Palermo* 48.1 (1999), pp. 163–190. ISSN: 1973-4409. DOI: 10.1007/BF02844387.

[86] M. E. McConney, N. Chen, D. Lu, H. A. Hu, S. Coombs, C. Liu, and V. V. Tsukruk. "Biologically inspired design of hydrogel-capped hair sensors for enhanced underwater flow detection." en. In: *Soft Matter* 5.2 (2009), pp. 292–295. DOI: 10.1039/B808839J.

[87] M. J. McHenry and S. M. van Netten. "The flexural stiffness of superficial neuromasts in the zebrafish (Danio rerio) lateral line." en. In: *Journal of Experimental Biology* 210.23 (2007), pp. 4244–4253. DOI: 10.1242/jeb.009290.

[88] P. van de Meulen, B. Wolf, P. Pirih, and S. van Netten. "Performance of Neural Networks in Source Localization using Artificial Lateral Line Sensor Configurations." In: Amersfoort: Poster presented at ICT.OPEN 2018, 2018. DOI: 10.13140/RG.2.2.34331.72481.

[89] J. C. Montgomery, C. F. Baker, and A. G. Carton. "The lateral line can mediate rheotaxis in fish." In: *Nature* 389.6654 (1997), p. 960. DOI: 10.1038/40135.

[90] H. Münz. "Functional organization of the lateral line periphery." In: *The mechanosensory lateral line*. Springer, 1989, pp. 285–297. DOI: 10.1007/978-1-4612-3560-6_14.

[91] M. Nakae, R. Asaoka, H. Wada, and K. Sasaki. "Fluorescent dye staining of neuromasts in live fishes: an aid to systematic studies." en. In: *Ichthyological Research* 59.3 (July 2012), pp. 286–290. ISSN: 1341-8998, 1616-3915. DOI: 10.1007/s10228-012-0274-2.

[92] S. M. van Netten. "Hydrodynamic detection by cupulae in a lateral line canal: functional relations between physics and physiology." en. In: *Biological Cybernetics* 94.1 (2006), pp. 67–85. DOI: 10.1007/s00422-005-0032-x.

[93] S. M. van Netten and M. J. McHenry. "The biophysics of the fish lateral line." In: *The lateral line system*. Ed. by S. Coombs, H. Bleckmann, R. R. Fay, and A. N. Popper. Vol. 48. New York, NY: Springer New York, 2013, pp. 99–119. DOI: 10.1007/2506_2013_14.

[94] A. Y. Ng. "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance." In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ICML '04. New York, NY, USA: ACM, 2004, p. 78. ISBN: 1-58113-838-5. DOI: 10.1145/1015330.1015435.

[95] N. Nguyen, D. Jones, S. Pandya, Y. Yang, N. Chen, C. Tucker, and C. Liu. "Biomimetic flow imaging with an artificial fish lateral line." In: *BIOSIGNALS 2008-1st International Conference on Bio-inspired Systems and Signal Processing*. 2008.

[96] S. Pandya, Y. Yang, D. L. Jones, J. Engel, and C. Liu. "Multisensor processing algorithms for underwater dipole localization and tracking using MEMS artificial lateral-line sensors." en. In: *EURASIP Journal on Advances in Signal Processing* 2006 (2006), pp. 1–9. DOI: 10.1155/ASP/2006/76593.

[97] S. Pandya, Y. Yang, C. Liu, and D. L. Jones. "Biomimetic imaging of flow phenomena." In: *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. Vol. 2. IEEE, 2007, pp. II–933. DOI: 10.1109/ICASSP.2007.366390.

[98] B. L. Partridge and T. J. Pitcher. "The sensory basis of fish schools: relative roles of lateral line and vision." In: *Journal of Comparative Physiology* 135.4 (1980), pp. 315–325. DOI: 10.1007/BF00657647.

[99] R. Pascanu, T. Mikolov, and Y. Bengio. "On the difficulty of training recurrent neural networks." In: *International Conference on Machine Learning*. 2013, pp. 1310–1318.

[100] M. A. Rapo, H. Jiang, M. A. Grosenbaugh, and S. Coombs. "Using computational fluid dynamics to calculate the stimulus to the lateral line of a fish in still water." en. In: *Journal of Experimental Biology* 212.10 (2009), pp. 1494–1505. DOI: 10.1242/jeb.026732.

[101] A. Ristolainen, K. Kalev, J. A. Tuhtan, A. Kuusik, and M. Kruusmaa. "Hydromorphological Classification Using Synchronous Pressure and Inertial Sensing." In: *IEEE Transactions on Geoscience and Remote Sensing* 56.6 (2018), pp. 3222–3232. ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2018.2795641.

[102] M. Robnik-Šikonja and I. Kononenko. "Theoretical and empirical analysis of ReliefF and RReliefF." In: *Machine learning* 53.1-2 (2003), pp. 23–69.

[103] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[104] O. Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge." In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[105] M Satou, H.-A. Takeuchi, J Nishii, M Tanabe, S Kitamura, N Okumoto, and M Iwata. "Behavioral and electrophysiological evidences that the lateral line is involved in the inter-sexual vibrational communication of the himé salmon (landlocked red salmon, Oncorhynchus nerka)." In: *Journal of Comparative Physiology A* 174.5 (1994), pp. 539–549. DOI: 10.1007/BF00217373.

[106] A. B. Sichert, R. Bamler, and J. L. van Hemmen. "Hydrodynamic Object Recognition: When Multipoles Count." en. In: *Physical Review Letters* 102.5 (2009). ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.102.058104.

[107] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." In: *arXiv preprint arXiv:1409.1556* (2014).

[108] S. W. Smith. "Chapter 17: custom filters." In: *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, 1997, pp. 297–310. URL: dspguide.com.

[109] S. K. Sønderby, C. K. Sønderby, H. Nielsen, and O. Winther. "Convolutional LSTM Networks for Subcellular Localization of Proteins." In: *Algorithms for Computational Biology*. Ed. by A.-H. Dediu, F. Hernández-Quiroz, C. Martín-Vide, and D. A. Rosenblueth. Cham: Springer International Publishing, 2015, pp. 68–80. ISBN: 978-3-319-21233-3. DOI: 10.1007/978-3-319-21233-3_6.

[110] G. G. Stokes. *On the effect of the internal friction of fluids on the motion of pendulums*. Vol. 9. Pitt Press Cambridge, 1851.

[111] N. Strokina, J.-K. Kamarainen, J. A. Tuhtan, J. F. Fuentes-Perez, and M. Kruusmaa. "Joint Estimation of Bulk Flow Velocity and Angle Using a Lateral Line Probe." In: *IEEE Transactions on Instrumentation and Measurement* 65.3 (2016), pp. 601–613. ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2015.2499019.

[112] A. Sutterlin and S. Waddy. "Possible role of the posterior lateral line in obstacle entrainment by brook trout (Salvelinus fontinalis)." In: *Journal of the Fisheries Board of Canada* 32.12 (1975), pp. 2441–2446. DOI: 10.1139/f75-281.

[113] S. Takashima, H. Asanuma, and H. Niitsuma. "A water flowmeter using dual fiber Bragg grating sensors and cross-correlation technique." en. In: *Sensors and Actuators A: Physical* 116.1 (2004), pp. 66–74. DOI: 10.1016/j.sna.2004.03.026.

[114] J. Tang, S. Alelyani, and H. Liu. "Feature selection for classification: A review." In: *Data classification: Algorithms and applications* (2014), p. 37.

[115] J. Tao and X. B. Yu. "Hair flow sensors: from bio-inspiration to bio-mimicking—a review." In: *Smart Materials and Structures* 21.11 (2012), p. 113001. DOI: 10.1088/0964-1726/21/11/113001.

[116] F. van Veen and S. Leijnen. *The Neural Network Zoo*. https://asimovinstitute.org/neural-network-zoo. 2019.

[117] A. N. Vollmayr, S. Sosnowski, S. Hirche, and L. J. van Hemmen. "Snookie: an autonomous underwater vehicle with Artificial Lateral Line system." In: *Flow Sensing in Air and Water*. Ed. by H. Bleckmann, J. Mogdans, and S. L. Coombs. 8th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 521–562. ISBN: 978-3-642-41445-9. DOI: 10.1007/978-3-642-41446-6_20.

[118] P. J. Werbos. "Applications of advances in nonlinear sensitivity analysis." In: *System Modeling and Optimization*. Ed. by R. F. Drenick and F. Kozin. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982, pp. 762–770. ISBN: 978-3-540-39459-4. DOI: 10.1007/BFb0006203.

[119] S. P. Windsor, S. E. Norris, S. M. Cameron, G. D. Mallinson, and J. C. Montgomery. "The flow fields involved in hydrodynamic imaging by blind Mexican cave fish (Astyanax fasciatus). Part I: open water and heading towards a wall." en. In: *Journal of Experimental Biology* 213.22 (Nov. 2010), pp. 3819–3831. ISSN: 0022-0949, 1477-9145. DOI: 10.1242/jeb.040741.

[120] S. P. Windsor and M. J. McHenry. "The influence of viscous hydrodynamics on the fish lateral-line system." In: *Integrative and comparative biology* 49.6 (2009), pp. 691–701. DOI: 10.1093/icb/icp084.

[121] B. J. Wolf and S. M. van Netten. "Hydrodynamic Imaging using an all-optical 2D Artificial Lateral Line." In: *2019 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2019, pp. 1–6. DOI: 10.1109/SAS.2019.8706030.

[122] B. J. Wolf and S. M. van Netten. "Training submerged source detection for a 2D fluid flow sensor array with extreme learning machines." In: *Eleventh International Conference on Machine Vision (ICMV 2018)*. Vol. 11041. International Society for Optics and Photonics. 2019, p. 1104126. DOI: 10.1117/12.2522667.

[123] B. J. Wolf, S. Warmelink, and S. M. van Netten. "Recurrent neural networks for hydrodynamic imaging using a 2D-sensitive artificial lateral line." In: *Bioinspiration & Biomimetics* 14.5 (2019), p. 055001. DOI: 10.1088/1748-3190/ab2cb3.

[124] B. J. Wolf, J. A. S. Morton, W. N. MacPherson, and S. M. van Netten. "Bioinspired all-optical artificial neuromast for 2D flow sensing." In: *Bioinspiration & Biomimetics* 13.2 (2018), p. 026013. ISSN: 1748-3190. DOI: 10.1088/1748-3190/aaa786.

[125] Y. Yang, J. Chen, J. Engel, S. Pandya, N. Chen, C. Tucker, S. Coombs, D. L. Jones, and C. Liu. "Distant touch hydrodynamic imaging with an artificial lateral line." In: *Proceedings of the National Academy of Sciences* 103.50 (2006), pp. 18891–18895. DOI: 10.1073/pnas.0609274103.

[126] Y. Yang, N. Chen, C. Tucker, J. Engel, S. Pandya, and C. Liu. "From artificial hair cell sensor to artificial lateral line system: development and application." In: *Micro Electro Mechanical Systems, 2007. MEMS. IEEE 20th International Conference on*. IEEE, 2007, pp. 577–580. DOI: 10.1109/MEMSYS.2007.4432986.

[127] Y. Yang, N. Nguyen, N. Chen, M. Lockwood, C. Tucker, H. Hu, H. Bleckmann, C. Liu, and D. L. Jones. "Artificial lateral line with biomimetic neuromasts to emulate fish sensing." In: *Bioinspiration & Biomimetics* 5.1 (2010), p. 016001. DOI: 10.1088/1748-3182/5/1/016001.

[128] Y. Yang, A. Klein, H. Bleckmann, and C. Liu. "Artificial lateral line canal for hydrodynamic detection." en. In: *Applied Physics Letters* 99.2 (2011), p. 023701. ISSN: 00036951. DOI: 10.1063/1.3610470.

[129] L. Yu and H. Liu. "Feature selection for high-dimensional data: A fast correlation-based filter solution." In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 856–863.

[130] M. D. Zeiler and R. Fergus. "Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I." In: ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014. Chap. Visualizing and Understanding Convolutional Networks, pp. 818–833. ISBN: 978-3-319-10590-1. DOI: 10.1007/978-3-319-10590-1_53.

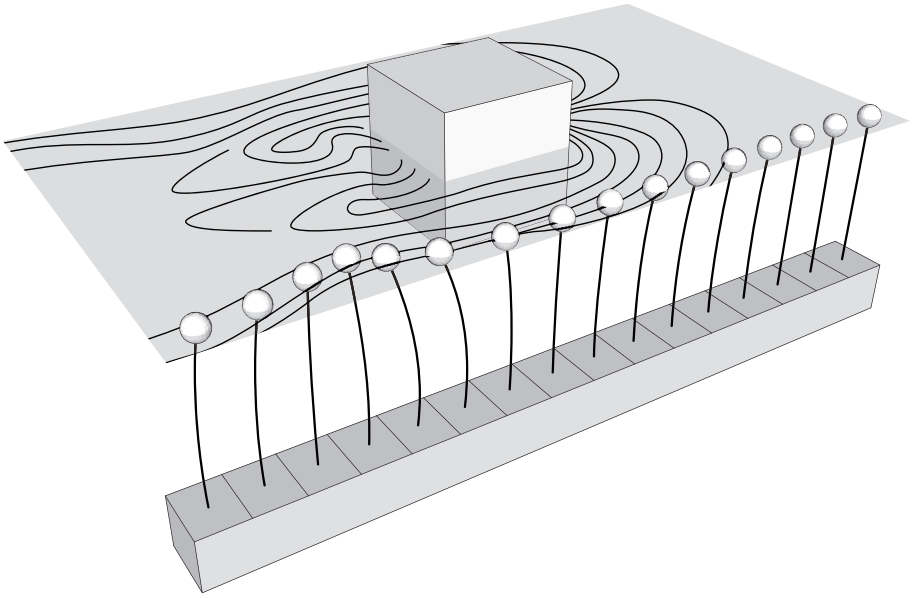
[131] X. Zheng, Y. Zhang, M. Ji, Y. Liu, X. Lin, J. Qiu, and G. Liu. "Underwater Positioning Based on an Artificial Lateral Line and a Generalized Regression Neural Network." In: *Journal of Bionic Engineering* 15.5 (2018), pp. 883–893. ISSN: 1672-6529, 2543-2141. DOI: 10.1007/s42235-018-0075-z.

[132] X. Zheng, C. Wang, R. Fan, and G. Xie. "Artificial lateral line based local sensing between two adjacent robotic fish." In: *Bioinspiration & Biomimetics* 13.1 (2017), p. 016002. ISSN: 1748-3190. DOI: 10.1088/1748-3190/aa8f2e.

NEDERLANDSE SAMENVATTING
VAN HET PROEFSCHRIFT

HYDRODYNAMISCHE PERCEPTIE *met* KUNSTMATIGE INTELLIGENTIE

Het op afstand detecteren van objecten in het water
met behulp van een 2D-gevoelige stromingssensor-array en neurale netwerken



object		sensor-array

• snelheid
• grootte
• vorm (§ 8)
• richting (§ 5)
• locatie
  • coördinaten (§ 3, 5, 6, 7)
  • waarschijnlijkheid (§ 4)
	vloeistofinteractie →
	via simulatie (II) of meting (III, IV)
	hydrodynamische perceptie ←
	via neurale netwerken
		• deflectie (§ 2)
		• vloeistofsnelheid, v
		  • parallel aan array, x
		  • loodrecht op array, y
		• dimensies voor metingen
		  • ruimte (§ 3, 4, 5, 6, 7)
		  • tijd (§ 4, 5, 6, 7, 8)

Vissen hebben een unieke kijk op hun omgeving onder water. Ze lijken zelden tegen de wand van een aquarium aan te botsen, ze kunnen in scholen zwemmen, en bovendien kunnen sommige vissen in donkere grotten of in de duistere diepte van oceanen tóch veilig navigeren. Naast de vijf zintuigen die wij als mensen kennen, beschikken zij over een zesde zintuig. In dit proefschrift bootsen we dit zintuig na en proberen we het te verbeteren.

Dit zesde zintuig wordt soms omschreven als *voelen op afstand*, en het wordt mogelijk gemaakt door het zijlijnorgaan. Dit orgaan bestaat uit stromingsdetectoren die de vloeistofsnelheid in de richting langs de zijlijn meten. Vissen kunnen met dit zintuig de nabije vloeistofstromingen interpreteren en op deze manier hun omgeving waarnemen. In dit proefschrift noemen we dit proces *hydrodynamische perceptie*. Dit stelt hen in staat om voedsel, prooidieren, en roofdieren op afstand te detecteren, en is ook belangrijk voor scholingsgedrag.

Het zijlijnorgaan heeft verscheidene onderzoekers geïnspireerd om sensorsystemen te bouwen die dit zesde zintuig kunnen nabootsen, een zogenoemde *kunstmatige* zijlijn (EN: *artificial lateral line*). Een kunstmatige zijlijn bestaat meestal uit een aantal sensoren naast elkaar; dit wordt ook wel een *sensor-array* genoemd. Deze systemen worden gebruikt om de nabije hydrodynamische omgeving te meten om obstakels te detecteren of om eigenschappen van de vloeistofstroming te bepalen, zoals hoe turbulent of hoe uniform die stroming is.

In dit proefschrift ontwikkelen we een nieuw type stromingssensor-array. We doen onderzoek naar de informatie die een bewegend object achterlaat in de stroming van het water én naar hoe we deze informatie kunnen gebruiken om eigenschappen van dat object te bepalen met behulp van kunstmatige intelligentie.

HYDRODYNAMISCHE PERCEPTIE

De afbeelding op de vorige pagina laat schematisch zien hoe een bewegend object door middel van vloeistofinteractie het water verplaatst en hoe deze stroming gemeten kan worden met een sensor-array. Het modelleren of simuleren van deze vloeistofinteractie tussen het bewegend object en de nabije vloeistof is niet eenvoudig: het beslaat een onderzoeksveld op zich. In de wis- en natuurkunde staat het uitzoeken van een dergelijk model bekend als een *voorwaarts* probleem (EN: *forward problem*). Het voorwaartse probleem is in dit geval om vast te stellen hoe de verschillende eigenschappen van een bron bijdragen aan een meetbare verandering in het nabije vloeistofveld.

Bij hydrodynamische perceptie redeneren we de andere kant op. We willen juist eigenschappen van de bron bepalen aan de hand van de gemeten stroming. Dit wordt omschreven als het *inverse* probleem (EN: *inverse problem*). Ook het oplossen van inverse problemen is een vakgebied op zich en is niet gelimiteerd tot stromingsmetingen. Ook toepassingen zoals CT-scans, sonar, en seismische metingen gebruiken een invers model om aan de hand van een verzameling meetpunten de samenstelling van een volume

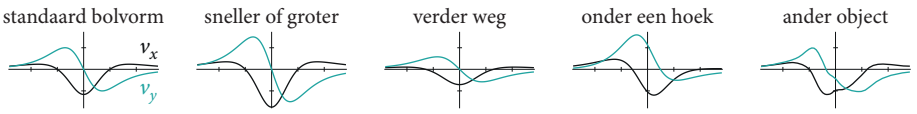
te bepalen of objecten te detecteren. Deze verschillende toepassingen zijn vaak wel gebaseerd op andere principes.

Hydrodynamische perceptie werkt in het zogenoemde nabije veld (EN: *near field*), wat betekent dat de afstand waarop je objecten kunt detecteren via hydrodynamische perceptie beperkt is ten opzichte van bijvoorbeeld sonar, dat werkt in het zogenoemde verre veld (EN: *far field*). Sonar en seismische metingen zijn immers gebaseerd op geluid, wat verder draagt en dus minder snel afzwakt: geluidssignalen nemen af met het kwadraat van de afstand, terwijl hydrodynamische signalen met de derde macht van de afstand afzwakken.

Informatie via vloeistofinteractie

Zoals eerder beschreven is het voorwaartse probleem om vast te stellen hoe de verschillende eigenschappen van de bron bijdragen aan een meetbare verandering in het nabije vloeistofveld. Wanneer we specifiek naar de aaneenschakeling van de metingen langs het sensor-array kijken, spreken we van het meten van een zogenoemd stromingsprofiel. Aan de hand van een vereenvoudigd voorwaarts model, kunnen we een aantal eigenschappen van de bron direct afleiden via dit gemeten stromingsprofiel.

We gebruiken in eerste instantie een vereenvoudigd vloeistofmodel, dat ten eerste aanneemt dat er geen turbulentie ontstaat. Ten tweede gaat dit vloeistofmodel er ook van uit dat het object bolvormig is. Hoe groter, sneller, en dichterbij het object is, des te sterker is het hydrodynamisch signaal. Maar het object beïnvloedt ook de gemeten vorm van het stromingsprofiel. Hierbij is zowel de relatieve locatie, als de bewegingsrichting, en de vorm van het object van belang.



Figuur NL.1: 2D-stromingsprofielen voor een aantal variaties van een object dat parallel beweegt ten opzichte van het sensor-array. De horizontale as beschrijft de meetpunten langs het array, de verticale as beschrijft de gemeten vloeistofsnelheid.

Als we kijken naar een bolvormig object dat parallel beweegt ten opzichte van het sensor-array, hebben de grootte en snelheid van de bron geen invloed op de vorm van de x- en y-stromingsprofielen (figuur NL.1), maar wel op de gemeten amplitude. Daarentegen heeft de afstand van de bron een iets ingewikkeldere relatie tot een gemeten stromingsprofiel. Wanneer het object dichterbij is, neemt de amplitude toe, en wordt het profiel ook smaller. Wanneer het object verder weg is, neemt de amplitude juist af, maar zien we ook een verbreding van het gemeten stromingsprofiel. Dit is bijvoorbeeld te zien aan de nulpunten van het profiel, die verder van het midden liggen. De vorm van dit profiel verandert niet met de afstand.

We beschouwen hier twee variaties die juist wel de vorm van een profiel beïnvloeden. Wanneer de bron in een andere richting beweegt, vervormen de x- en y-stromingsprofielen. Het x-profiel is dan bijvoorbeeld niet langer symmetrisch. Als laatste hebben we dan nog de vorm van het object. Deze heeft ook een invloed op de vorm van het stromingsprofiel. Maar wanneer de afstand tussen de bron en het sensor-array groter wordt, verkleind dit effect en lijkt elk object op een bol.

In deel II van het proefschrift maken we gebruik van dit vereenvoudigde model voor het simuleren van een bewegende bol. Allereerst helpt deze simulatie ons om te bepalen of hydrodynamische perceptie überhaupt mogelijk is met neurale netwerken. Daarnaast stelt ze ons ook in staat om kwalitatief objectieve vergelijkingen te maken tussen verschillende inverse methoden. We kunnen bovendien gemakkelijk variaties van sensor-arrays uitproberen. Zo kunnen we in een simulatie het aantal metingen kiezen of de gevoeligheid en het aantal sensoren aanpassen, maar is dit in een testopstelling niet eenvoudig is; in sommige gevallen is het zelfs niet mogelijk.

Een simulatiestudie gebaseerd op het eerdergenoemde vereenvoudigde vloeistofmodel kent ook zijn limieten. Een dergelijk model is gebaseerd op aannames die alleen standhouden wanneer de snelheden van de objecten en het water beperkt zijn. Daarom is het ook nodig om echte metingen uit te voeren om vast te stellen of hydrodynamische perceptie niet alleen werkt in een simulatie, maar ook in de praktijk. Daarom bestaan deel III en IV van het proefschrift uit experimenten met metingen van stromingsprofielen van trillende bollen, bewegende bollen, en bewegende objecten met een andere vorm.

Hydrodynamische perceptietaken

Bij hydrodynamische perceptie onderscheiden we in dit proefschrift een aantal deeltaken, die elk als doel hebben het eerder beschreven inverse probleem deels op te lossen. Concreet willen we aan de hand van een gemeten stromingsprofiel een aantal eigenschappen achterhalen van de bron, zoals de locatie, bewegingsrichting, en de vorm. Binnen dit proefschrift hebben we kwalitatief vier verschillende hydrodynamische perceptietaken gedaan met behulp van kunstmatige intelligentie:

- het bepalen van de locatie van een object (hoofdstukken 3, 5, 6, en 7);
- het bepalen van de bewegingsrichting van een object (hoofdstuk 5);
- het bepalen van de vorm van een bewegend object (hoofdstuk 8);
- en het bepalen van de waarschijnlijkheid dat een of meerdere bronnen aanwezig zijn in een bepaald volume (hoofdstuk 4).

BELANGRIJKSTE RESULTATEN

In de vorige sectie hebben we kort beschreven hoe een object zijn vloeistofomgeving kan beïnvloeden via vloeistofinteractie, gevolgd door een beschrijving van vier verschillende deeltaken van hydrodynamische perceptie.

Ten opzichte van de bestaande literatuur over hydrodynamische perceptie kunnen we de nieuwe bijdragen van dit proefschrift aan het onderzoeksveld relateren aan drie thema's: schaalbaarheid, signaalverwerking met neurale netwerken, en metingen van 2D-stromingsprofielen. In de volgende secties vatten we de belangrijkste resultaten samen voor elk van deze thema's.

Schaalbaarheid van de technologie en de toepassing

In deel I beschrijven we het ontwerp van een nieuw type stromingssensor en het bijbehorende mechanische model. Op de afbeelding aan het begin van deze samenvatting (pagina 191) zien we een array van deze sensoren in een opstelling. Elke sensor bestaat uit een buigzame optische glasvezelstructuur en een kleine (witte) bol die de vloeistofkrachten oppikt.

Deze volledig optische stromingssensoren werken geheel passief; ze hebben geen elektrische bron nodig om te functioneren. In plaats daarvan werken ze via zogenoemde Bragg-rasters; deze zijn aangebracht in de glasvezels en maken het mogelijk om de buiging van de sensoren te meten. Wanneer deze sensoren gekoppeld zijn aan een laserbron, kunnen we aan de hand van de lichtreflectie door het Bragg-raster precies bepalen hoe de sensor ombuigt. Een groot voordeel van dit systeem is dat de sensoren geen lokale elektriciteit nodig hebben, en daarom op kilometers afstand kunnen worden uitgelezen. Bovendien bestaat de sensor zelf uit roestvrije materialen: glasvezel en kunststof. Deze eigenschappen maken dit type sensor uitermate geschikt voor langdurige en afgelegen stromingsmetingen in een zout of agressief milieu.

Met het mechanische model kunnen we dynamische eigenschappen van de sensor beschrijven zoals de gevoeligheid van de sensor voor verschillende frequenties. In hoofdstuk 2 leggen we uit hoe de vloeistofstroming omgezet wordt in een sensorbeweging en hoe het uiteindelijke sensorsignaal eruit ziet. Via een aantal kalibratiemetingen laten we zien dat het mechanische model een goede beschrijving is van de sensor. Dit stelt ons in staat om de verschillende dimensies in het sensorontwerp te optimaliseren om zoveel mogelijk gemeten signaal per vloeistofsnelheid op te wekken.

Naast het schaalbaar maken van de achterliggende technologie, onderzoeken we ook of hydrodynamische perceptie toegepast kan worden op een grotere schaal dan door vissen. In deel III en IV van dit proefschrift schalen we daarom het kunstmatige zijlijnorgaan op in enkele stappen en laten zien dat de hydrodynamische perceptietaken succesvol uitgevoerd kunnen worden. Zie ook tabel NL.1 voor een overzicht.

Tabel NL.1: Samenvatting van de perceptietaken elk sensor-array zoals gebruikt in de hoofdstukken (§) die experimenten beschrijven.

§	TAAK	AANTAL SENSOREN	LENGTE ARRAY	DETECTIEGEBIED
6	locatie	4	16 cm	36 × 11 cm
7	locatie	8	45 cm	62 × 9,5 cm
8	vorm	8	350 cm	~ 600 × 100 cm

We beginnen met vier sensoren in een array van 16 cm om zowel een trillende als een bewegende bron te lokaliseren. In een gebied van 36 × 11 cm komen we tot een gemiddelde schattingsfout van 1,3 cm voor de trillende bron en 3,3 cm voor een bron die in een rechte lijn beweegt. We schalen dit verder op naar een array van 45 cm en acht sensoren. In een gebied van 62 × 9,5 cm kunnen we met dit langere array een bewegend object lokaliseren met een schattingsfout van 0,72 cm.

In hoofdstuk 8 laten we een andere hydrodynamische perceptietaak zien. Met acht sensoren in een array van 3,5 meter proberen we de vorm van een voorbijgaand object te bepalen aan de hand van de vloeistofstroming gemeten door het array. Dit is de eerste keer dat vormherkenning op deze schaal en met dit aantal (vijf) verschillende objecten wordt gedemonstreerd. Met ons geïntroduceerde classificatiealgoritme kunnen we de vorm van het voorbijgaand object voorspellen met een nauwkeurigheidsscore (F_1) van 98%.

Kortom, zowel de gebruikte technologie als hydrodynamische perceptie blijkt behoorlijk opgeschaald te kunnen worden ten opzichte van de biologisch voorkomende afmetingen. Er zijn daarbij wel limieten aan de afstanden waarop bronnen kunnen worden gedetecteerd. Omdat het sensorsysteem passief is, hangt de detecteerbare afstand vooral af van de bron: hoe groter en sneller de bron, hoe meer water wordt verplaatst. Hierdoor wordt het hydrodynamische signaal sterker en dus de maximale detectieafstand groter.

Signaalverwerking met neurale netwerken

In dit proefschrift zijn voornamelijk kunstmatige neurale netwerken gebruikt om de hydrodynamische perceptietaken uit te voeren. De vele verschillende types neurale netwerken hebben elk hun eigen karakteristieke eigenschappen die hen beter geschikt maken voor een bepaalde toepassing; zie ook de afbeelding in de introductie op pagina 8. We experimenteren daarom met verschillende types neurale netwerken om te bepalen welk type neuraal netwerk of signaalverwerking optimaal is voor de vier genoemde hydrodynamische perceptietaken. Deze zijn ook verderop aangeduid in tabel NL.2.

Hoofdstuk 9 bevat een uitgebreide discussie over de verschillende types netwerk en signaalverwerking zoals gebruikt in de hoofdstukken 3 tot en met 8. We vatten hier de hoofdpunten samen. We hebben verschillende soorten neurale netwerken vergeleken, met als standaard een eenvoudig maar krachtig type: de ELM. Deze werd vergeleken met de MLP, het ESN, en de LSTM. Bovendien werd het CNN ook gebruikt in hoofdstuk 4 om een beeld te maken van de omgeving. Al deze netwerken, mits geoptimaliseerd om *overfitting* te voorkomen, zijn aantoonbaar geschikt om één van de hydrodynamische perceptietaken uit te voeren.

Voor elk van de vier perceptietaken, kan een andere manier van dataverwerking optimaal zijn. Hierbij maken we onderscheid tussen hoe de spatiële dimensie (ruimte) en temporele dimensie (tijd) van de gemeten signalen verwerkt wordt.

LOCATIE Wanneer we een trillende bron moeten lokaliseren (hoofdstukken 5 en 6), of een bron die onvoorspelbaar beweegt (hoofdstukken 3 en 4), zit de meeste informatie in de spatiële dimensie: de meetpunten van elke sensor in het array. De temporele dimensie wordt in deze gevallen voornamelijk gebruikt om de signaal-ruiskwaliteit van het signaal te verbeteren. Zeker wanneer het object een vaste trillingsfrequentie heeft, kan de kwaliteit van het spatiële stromingsprofiel via een zogenoemde Fouriertransformatie merkbaar verbeterd worden.

Wanneer de beweging van de bron meer voorspelbaar is (hoofdstuk 5), of zelfs een rechte lijn volgt (hoofdstukken 6 en 7), dan zijn zowel de spatiële als de temporele dimensie van de metingen van belang. Er zijn twee manieren om de informatie van de temporele dimensie te gebruiken. De impliciete manier is door het gebruiken van neurale netwerken met een interne dynamiek, zoals bij de ESN en het LSTM-netwerk. Deze interne dynamiek zorgt voor een soort feedback over de tijd. Dit houdt in dat de uitvoer van het netwerk niet alleen gebaseerd is op de huidige invoer, maar ook afhankelijk is van de informatie die daarvoor aan het neurale netwerk gevoerd is. De interne representatie van de geschiedenis is in dit geval niet direct inzichtelijk. De andere manier is het expliciet toevoegen van geschiedenis door meerdere stromingsprofielen als invoer te geven; in plaats van alleen het huidige stromingsprofiel, bieden we een klein aantal voorafgaande metingen aan. Deze expliciete manier kan worden toegepast bij elk type neuraal netwerk.

Voor het bepalen van de locatie is dus de spatiële dimensie in elk geval van belang. Afhankelijk van de het type bronbeweging, kan het in sommige gevallen beter zijn om de temporele dimensie te gebruiken om de kwaliteit van het spatiele profiel te verbeteren, maar zijn in de meeste gevallen beide van belang.

BEWEGINGSRICHTING We hebben alleen in simulatie gekeken naar het bepalen van de bewegingsrichting van een object. Hoewel de bewegingsrichting al gecodeerd zit in de spatiele dimensie, namelijk het stromingsprofiel (zie afbeelding NL.1), is het voordelig

om meerdere tijdpunten te beschouwen om het traject en op die manier de richting van de bron te bepalen.

VORM In hoofdstuk 8 bepalen we de vorm van een object aan de hand van het temporele stromingsprofiel. Omdat het array flink is opgeschaald zonder het aantal sensoren te vergroten, is de spatiële resolutie te laag om direct de vorm van het object goed te bepalen. In dit geval wordt de spatiële informatie gecombineerd of gemiddeld tot een paar kenmerkende waarden (EN: *features*) per tijdstap.

AANWEZIGHEID In hoofdstuk 4 voeren we een geheel ander type taak uit. Voor de vorige taken reduceren we de stromingsprofielen tot een aantal waarden, zoals de coördinaten van de bron. Voor de huidige taak proberen we met een convolutioneel neuraal netwerk (CNN) juist een grote hoeveelheid waarden te voorspellen: een volledig zogenoemde hydrodynamische afbeelding. Deze hydrodynamische afbeelding geeft voor elke coördinaat in 2D de waarschijnlijkheid aan of zich hier een bron bevindt. We simuleren twee arrays om via twee hydrodynamische afbeeldingen de locatie in 3D te vinden van één of meerdere bronnen. Bij deze toepassing lijkt het expliciet toevoegen van temporele informatie niet veel te helpen, maar er is nog ruimte voor experimentele variaties die iets anders kunnen uitwijzen.

Tabel NL.2: Uitgebreide samenvatting van de perceptietaken en elk type sensor. Voor elk van de hoofdstukken (§) die simulaties (sim) en experimenten (exp) beschrijven, geven we ook het type dataverwerking aan.

§	TAAK	SENSOREN	TYPE BRON	TYPE NN	TYPE INVOER
3	locatie	1D sim	bewegend	MLP, ELM, ESN	spatieel
4	aanwezigheid	1D sim	bewegend	CNN	beide
5	locatie & richting	1D sim & 2D sim	trillend & bewegend	ELM	beide
6	locatie	2D exp	trillend & bewegend	ELM	beide
7	locatie	2D exp	bewegend	OS-ELM, LSTM	beide
8	vorm	2D exp	bewegend	ELM	temporeel

Over het geheel genomen lijkt het optimaal om zowel de temporele als de spatiële dimensie van de meting te gebruiken. Dit kan zijn om het signaal-ruisniveau te verbeteren of om extra zinvolle informatie toe te voegen voor de uiteindelijke schatting.

Uitbreiding naar het meten van 2D-stromingsprofielen

In dit proefschrift laten we zien dat we verschillende taken succesvol kunnen uitvoeren met de 2D-gevoelige sensoren van hoofdstuk 2. In de (kunstmatige) zijlijnliteratuur wordt vooral gewerkt met 1D-stromingsprofielen. Ten opzichte van deze profielen zien we dat de nieuwe y-component van de vloeistofsnelheid een complementair effect heeft. Deze nieuwe component staat loodrecht op het array en wordt in de biologische zijlijn door de vis niet benut.

Het complementaire effect is het duidelijkst zichtbaar in de stromingsprofielen zoals in figuur NL.1. Wanneer de ene component een nulpunt heeft in het stromingsprofiel, heeft de ander vaak een maximum of minimum. Dit houdt in dat er op deze speciale punten een aanzienlijke verbetering is van de signaal-ruisverhouding. Dit ligt in lijn met onze observatie dat de bepalingen op basis van de 2D stromingsprofielen beter bestand zijn tegen ruis dan wanneer traditionele 1D profielen gebruikt worden. Bovendien wordt het detectiegebied dat correct wordt gemeten door de sensoren daardoor ook groter, voornamelijk in de breedte. Zie ook de resultaten van de experimenten in hoofdstukken 6 en 7.

Doordat we in hoofdstuk 8 kijken naar verschillende combinaties van zogenoemde stromingskenmerken (EN: *flow features*), wordt ook duidelijk dat het combineren van deze twee vloeistofcomponenten tot de beste resultaten leidt voor vormherkenning. Uit deze studie wordt ook duidelijk dat de nieuwe y-component op zichzelf tot een betere vormherkenning leidt dan de parallelle x-component, zoals gemeten door de meeste kunstmatige zijlijnorganen in het onderzoeksveld.

Al met al zijn sensoren die beide meetrichtingen combineren uitermate geschikt voor het uitvoeren van hydrodynamische perceptietaken, zoals het lokaliseren van een object en het bepalen van de vorm.

CONCLUSIE

In dit proefschrift wordt vooral gefocust op het bepalen van de locatie, de richting, en de vorm van een nabij object. Er zijn verschillende vervolgstappen en toepassingen denkbaar voor de ontwikkelde technologie. Dit is een passieve (en daardoor ondetecteerbare) methode om nabije objecten in het water te detecteren en te herkennen. Het meetprincipe werkt bovendien in troebel water én in het donker; dit is een voordeel, want dat is iets wat sonar- en camerabeelden nog wel eens kan dwarszitten.

Zo kan hydrodynamische perceptie via kunstmatige intelligentie een toevoeging zijn op het bijhouden van waterverkeer in waterwegen en havens, al is het detectiegebied

van deze technologie wel kleiner ten opzichte van sonar en camera's. Omgekeerd zou de technologie gebruikt kunnen worden voor veilige navigatie wanneer we het toepassen op een schip. Juist omdat sonar op de zeer korte afstanden minder betrouwbaar is, kan deze technologie helpen om een beter beeld te vormen van de omgeving. Vissen gebruiken het zijlijnorgaan bijvoorbeeld ook om nabije wanden en andere obstakels te detecteren, maar deze toepassing laat zich minder eenvoudig nabootsen; de eigen boeggolven en trillingen van een vaartuig kunnen gemakkelijk interfereren met het hydrodynamisch signaal.

We hebben in dit proefschrift veelbelovende eerste stappen gezet voor nieuwe toepassingen van hydrodynamische perceptie. De nieuwe bijdragen kunnen worden samengevat in drie thema's. We hebben aangetoond dat hydrodynamische perceptie aanzienlijk kan worden opgeschaald, dat neurale netwerken zeer geschikt zijn voor deze hydrodynamische perceptietaken, en dat de uitbreiding naar 2D-sensoren de prestaties over het algemeen verbetert.

LIST OF PUBLICATIONS

PATENT APPLICATION

S.M. van Netten, B.J. Wolf, W.N. MacPherson, (2017). Sensor element and method for measuring of near-field, large-scale hydrodynamic characteristics. *EP3399320 (A1)*.

JOURNAL ARTICLES

B.J. Wolf, J. van de Wolfshaar, S.M. van Netten, (2020). Three-dimensional multi-source localization of underwater objects using convolutional neural networks for artificial lateral lines. *Journal of The Royal Society Interface* 17(162) 20190616.

B.J. Wolf, P. Pirih, M. Kruusmaa, S.M. van Netten, (2020). Shape classification using hydrodynamic detection via a sparse large-scale 2D-sensitive artificial lateral line. *IEEE Access* 8 11393-11404.

B.J. Wolf, S. Warmelink, S.M. van Netten, (2019). Recurrent neural networks for hydrodynamic imaging using a 2D-sensitive artificial lateral line. *Bioinspiration & Biomimetics* 14(5) 055001.

B.J. Wolf, J.A.S. Morton, W.M. macPherson, S.M. van Netten, (2018). Bio-inspired all-optical artificial neuromast for 2D flow sensing. *Bioinspiration & Biomimetics* 13(2) 026013.

L.H. Boulogne, B.J. Wolf, M.A. Wiering, S.M. van Netten, (2017). Performance of neural networks for localizing moving objects with an artificial lateral line. *Bioinspiration & Biomimetics* 12(5) 056009.

ARTICLES IN PREPARATION

B.J. Wolf, S.M. van Netten. Biophysics of the lateral line and applications. *submitted as chapter 7.6 of The Senses, 2nd Edition*.

CONFERENCE PROCEEDINGS

B.J. Wolf, S.M. van Netten, (2019). Hydrodynamic imaging using an all-optical 2D artificial lateral line. *2019 IEEE Sensors Applications Symposium (SAS)* 1-6.

B.J. Wolf, S.M. van Netten, (2019). Training submerged source detection for a 2D fluid flow sensor array with extreme learning machines. Proc. SPIE 11041, *Eleventh International Conference on Machine Vision (ICMV 2018)* 1104126.

POSTERS

B.J. Wolf, (2019). Tracking water flow and water traffic without sight and sound: combining AI and novel fluid flow sensing. *Meet and greet XL: Digital society*.

P. van der Meulen, B.J. Wolf, P. Pirih, S.M. van Netten, (2018). Performance of neural networks in source localization using artificial lateral line sensor configurations. *ICT.OPEN2018*.

DELIVERABLES

B.J. Wolf, S.M. van Netten, (2018). Demonstration of integrated canal based fibre optical array. *Deliverable 1.3 for H2020 Lakhsmi consortium*.

P. Pirih, B.J. Wolf, S.M. van Netten, (2018). Development of array hydrodynamic imaging methods. *Deliverable 3.5 for H2020 Lakhsmi consortium*.

B.J. Wolf, S.M. van Netten, (2017). Demonstration of array detection capabilities for imaging. *Deliverable 1.2 for H2020 Lakhsmi consortium*.

B.J. Wolf, S.M. van Netten, (2016). Sensor physics for basic sensor unit. *Deliverable 1.1 for H2020 Lakhsmi consortium*.

ACKNOWLEDGMENTS

This thesis is the culmination of a four-year journey, where very few steps were taken alone. I wouldn't have guessed that the first steps would lead me to a barn roughly 10 kilometers from Groningen, to retrieve high-end lab equipment from storage to kick start the first experiments.

In the next years, I got to visit and collaborate with the partners of the H2020 Lakhsmi project and other departments within Groningen. This includes the sensor deployments on Orkney, the sensor construction at Heriot Watt University, the experiments at the Biorobotics lab at the Technical University of Tallinn, the experiments performed in the University of Groningen Ocean Ecosystems labs, and those at the Hanze Academy of Sport Sciences. I've had the pleasure to meet and work with many people to think and talk about this interdisciplinary work involving engineering, machine learning, and hydrodynamics.

During this journey, my supervisors played an important role. I am very grateful for the opportunity provided by Sietse van Netten to work on this project as a PhD-student. His boundless knowledge, interest, enthusiasm, and optimism for the subject made for a very enjoyable and a very welcoming working environment. Whereas Sietse mentored me in the content of the project, Lambert Schomaker provided the necessary context. As my promotor, Lambert pushed me to consider the bigger picture with respect to demonstrating artificial intelligence in the real world.

I am also grateful that many students have entrusted me with supervising their final projects. Besides consuming the information our small group provided, they have contributed to the research in our group which in some cases led to a publication. Luuk, Robin, Li, Siger, Jordi, Pytrik, Eline, Arjen, Folke, Jelle, Ziyu, Jonathan, Christiaan, Ingeborg, Stijn, Wester, Hidde, Niek, and Anco; I have enjoyed supervising your bachelor projects. Pim, Steven, Sybren, Jelmer, and Lucia; thank you for sticking around a little longer for your masters project. I have learned a great deal from your contributions.

I would also like to thank Xiaoyin, Primoz, and Stephen for being warm and helpful office mates during the start, body, and end of my PhD respectively. My time here would have also been a lot less enjoyable without the many lunch and coffee moments with the academic staff, support staff, RoboCup team members, and fellow PhD students of the former Alice and current Bernoulli institute.

And lastly, I thank my friends and family for their support while simultaneously making sure that I kept my feet on the ground. As for my friends from the study association Cover, both inside and outside the many committees and boards I've been a part of: it has been my pleasure to have studied, worked, socialized, and networked with you.